(18) ARO 1110

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (19) 15561.6-M; 16835.2-M	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Comparison of Numerical Methods for Identification and Optimization Problems Involving Control Systems with Delays		5. TYPE OF REPORT & PERIOD COVERED Technical rept.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) H. T. /Banks J. A. /Burns E. M. /Cliff		8. CONTRACT OR GRANT NUMBER(s) DAAG29-78-G-0125 DAAG29-79-C-0161 (15)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Brown University Providence, Rhode Island 02912		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office P. O. Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE Nov 79
		13. NUMBER OF PAGES 185
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

LEVEL

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

(12) 189

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

differential equations
differential control systems
numerical methods
optimization problems

control systems
optimal control problems
estimation problems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Numerical results are presented for two approximation techniques for functional differential control systems. One technique is based on an averaging scheme, the other on spline approximations. A number of examples are considered and the techniques are applied to parameter estimation problems and optimal control problems where the systems are given by differential equations with hereditary terms.

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE

TR 79-7

H.T. BANKS
J.A. BURNS
E.M. CLIFF

A COMPARISON OF NUMERICAL METHODS FOR
IDENTIFICATION AND OPTIMIZATION PROBLEMS
INVOLVING CONTROL SYSTEMS WITH DELAYS

LEVEL

NOVEMBER 1979

AD A081986

Lefschetz Center for Dynamical Systems

A Comparison of Numerical Methods

for

Identification and Optimization Problems

Involving Control Systems With Delays

(November, 1979)

H. T. Banks*

Lefschetz Center for Dynamical Systems
Division of Applied Mathematics
Brown University
Providence, RI 02912

J. A. Burns**

Department of Mathematics
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

E. M. Cliff**

Aerospace and Ocean Engineering Department
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

A COMPARISON OF NUMERICAL METHODS FOR IDENTIFICATION

AND OPTIMIZATION PROBLEMS INVOLVING CONTROL SYSTEMS WITH DELAYS

H.T. Banks, J.A. Burns and E.M. Cliff

ABSTRACT

In this report we present numerical results for two approximation techniques for functional differential control systems. One technique is based on an averaging scheme, the other on spline approximations. A number of examples are considered and the techniques are applied to parameter estimation problems and optimal control problems where the systems are given by differential equations with hereditary terms.

1. Introduction.

This report is devoted to a detailed study, via numerical experiments, of certain algorithms for the control and identification of linear hereditary systems. In particular, we shall consider two schemes ("averaging" and linear "spline" approximations) which are based on the general approximation method developed in [1], [2], [3], [5]. The purpose of this report is to present examples which facilitate a comparison of the numerical performance of these two schemes for parameter identification and for optimal control problems. We shall not dwell upon theoretical convergence results. The interested reader is referred to [5] for complete statements of convergence results, error estimates and detailed proofs.

2. Notation and problem formulation.

The following notation will be used throughout the paper. For $-\infty < a < b < +\infty$, $L_p(a,b;R^n)$ is the usual Lebesque space of equivalence classes of all functions $x:[a,b] \to R^n$ such that $|x|^p$ is integrable. Let $H = [0,r]$ where $r > 0$ is a fixed real number, and Z will denote the Hilbert space $Z = R^n \times L_2(-r,0;R^n)$. For $0 < \tau \le r$, the space $R^n \times L_2(-\tau,0;R^n)$ will be denoted by $Z(\tau)$. The Sobolev space $W_2^{(1)}(-r,0;R^n)$ consists of all functions in $L_2(-r,0;R^n)$ with derivatives also in $L_2(-r,0)$ and norm given by

$$|\varphi|^2_{W_2^{(1)}} = |\varphi|^2_{L_2} + |\dot{\varphi}|^2_{L_2}.$$ We assume that Ω and S are compact convex subsets of R^μ and Z, respectively. Moreover, S is assumed to have the property that if $\varphi \in S$ and $0 < \tau < r$, then the projected function

$$(2.0) \qquad \widetilde{\varphi}(s) = \begin{cases} \varphi(s) & , \quad -\tau \leq s \leq 0 , \\[2em] 0 & , \quad -r \leq s < -\tau , \end{cases}$$

also belongs to S.

Let $Q = \Omega \times \mathcal{H}$ and $\Gamma = S \times Q$, so that a generic element of Γ has the form $\gamma = (\eta, \varphi, q) = (\eta, \varphi, \alpha, \tau)$. The elements γ of Γ are called the <u>system</u> <u>parameters</u>. We assume that for each $\alpha \in \Omega$, $A_0(\alpha)$, $A_1(\alpha)$ belong to $R^{n \times n}$, $B(\alpha) \in R^{n \times m}$, $C(\alpha) \in R^{k \times n}$, $D(\alpha) \in R^{k \times m}$ and $K(\alpha, \cdot)$ is an $n \times n$ matrix valued function with columns in $L_2(-r, 0; R^n)$.

If $x: [-\tau, +\infty) \to R^n$ and $t \geq 0$, the function $x_t: [-\tau, 0] \to R^n$ is defined by $x_t(s) = x(t+s)$. For $q = (\alpha, \tau) \in Q$, the operator $L(q)$ is defined by

$$L(q)\varphi = A_0(\alpha)\varphi(0) + A_1(\alpha)\varphi(-\tau) + \int_{-\tau}^{0} K(\alpha, s)\varphi(s)\, ds.$$

We consider the system governed by the linear retarded functional differential equation

$$(2.1) \qquad \dot{x}(t) = L(q)x_t + B(\alpha)u(t), \quad t \geq 0,$$

with initial data

$$(2.2) \qquad x(0) = \eta \quad , \quad x_0 = \varphi ,$$

and output

$$(2.3) \qquad y(t) = C(\alpha)x(t) + D(\alpha)u(t) \quad ,$$

where u is a R^m — valued, locally integrable control function and $(\eta,\varphi) \in Z$. Given a control u and $\gamma \in \Gamma$, the solution of the initial value problem (2.1)-(2.2) at time t will be denoted by $x(t;\gamma,u)$. The corresponding output to (2.1)-(2.3) at time $t \geq 0$ will be denoted by $y(t;\gamma,u)$.

REMARK. Note that the initial function φ need only be defined on $[-\tau,0]$. If $\tau < r$ and $\varphi \in L_2(-\tau,0;R^n)$ we shall identify φ with the projected function $\widetilde{\varphi} \in L_2(-r,0;R^n)$ defined by (2.0). With this understanding, any function in $L_2(-\tau,0;R^n)$ is also an element of $L_2(-r,0;R^n)$. Consequently, for notational convenience we shall not distinguish between $\varphi \in L_2(-\tau,0;R^n)$ and $\widetilde{\varphi} \in L_2(-r,0;R^n)$.

We shall be concerned with the system on a fixed finite interval $[0,T]$ where $T > 0$. The matrices G and W are n x n symmetric positive semi-definite, R is an m x m symmetric positive definite matrix and $\overline{y}_1$, $\overline{y}_2$, $\cdots$, $\overline{y}_M$ are given "observations" in R^k at times t_i, $0 \leq t_1 < t_2 < \cdots < t_M \leq T$. The above notation is summarized in the following list of nomenclature:

$\Gamma = S \times Q = S \times \Omega \times \aleph$ —   Parameter set

$\gamma = (\eta,\varphi,q) = (\eta,\varphi,\alpha,\tau)$ —   System parameters

G, W ----------------------------- Symm. positive semi-definite

R ------------------------------- Symm. positive definite

T ------------------------------- Fixed final time

$0 \le t_1 < \cdots < t_M \le T$ ------------ Observation times

$\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_M$ ----------------- Observations.

We now formulate two infinite dimensional optimization problems associated with the hereditary system described above.

The identification problem may be stated as follows:

PROBLEM (ID). Given the control u in $L_2(0,T;R^m)$ and observations $\bar{y}_i \in R^k$ at times t_i, find the system parameters $\gamma^* \in \Gamma$ which minimizes the fit error

$$(2.4) \qquad E(\gamma) = \tfrac{1}{2} \sum_{i=1}^{M} |y(t_i;\gamma,u) - \bar{y}_i|^2 ,$$

where $y(t;\gamma,u)$ is the output to (2.1)-(2.3), and the minimization takes place over Γ.

The optimal control problem may be stated as follows:

PROBLEM (OC). Given the system parameters $\gamma \in \Gamma$, find a control u^* in $L_2(0,T;R^m)$ which minimizes the performance criterion

$$(2.5) \qquad J(u) = \tfrac{1}{2}[x^T(T)Gx(T)] + \tfrac{1}{2} \int_0^T \{x^T(s)Wx(s) + u^T(s)Ru(s)\}\,ds,$$

where $x(t) = x(t;\gamma,u)$ is the solution to the system (2.1)-(2.2). The optimal cost will be denoted by J^* (i.e. $J^* = J(u^*)$).

It should be noted that the above formulation of the identification problem allows for the case where some of the system parameters $\gamma = (\eta, \varphi, \alpha, \tau)$ are known.

3. <u>The</u> <u>abstract</u> <u>Cauchy</u> <u>problem</u> <u>and</u> <u>approximations</u>.

In order to implement any numerical algorithm for solving the identification and optimal control problems, it is necessary to introduce approximations at some stage of the solution process. The basic idea used in this paper is to approximate the hereditary system by an ordinary differential system. We give a brief outline of the general framework and present two particular schemes. Details of the method may be found in [2], [3] and [5].

It is helpful to formulate the hereditary system (2.1)-(2.3) as an abstract system in the Hilbert space Z. Although this formulation is not essential if one is concerned only with numerical results, it is informative and indeed necessary if one is to fully understand the basic ideas underlining the methods to be discussed here.

Given $q = (\alpha, \tau) \in \Omega \times \mathcal{H}$, define for $t \geq 0$ the mapping $S(t;q):Z(\tau) \to Z(\tau)$ by

$$S(t;q)(\eta, \varphi) = (x(t;q), x_t(\cdot;q))$$

where $x(\cdot;q)$ is the solution to the homogeneous equation $\dot{x}(t) = L(q)x_t$ with initial condition $(x(0), x_0) = (\eta, \varphi)$. It is well known that for each fixed $q, \{S(t;q)\}_{t \geq 0}$ is a C_0-semigroup on $Z(\tau)$ (see [2], [3]). Moreover, the infinitesimal generator of $\{S(t,q)\}_{t \geq 0}$ is the operator $\mathcal{Q}(q)$ defined on the domain

$$\mathcal{D}(\mathcal{Q}(q)) = \{(\eta, \varphi) \in Z(\tau) \mid \varphi \in W_2^{(1)}(-\tau, 0; R^n), \varphi(0) = \eta\}$$

by

$$\alpha(q)(\eta,\varphi) = (L(q)\varphi,\dot{\varphi}) \quad .$$

Define $\beta(q):R^m \to Z(\tau)$ and $C(q):Z(\tau) \to R^k$ by $\beta(q)u = (B(\alpha)u,0)$ and $C(q)(\eta,\varphi) = C(\alpha)\eta$, respectively. Corresponding to the hereditary system (2.1)-(2.3) we have the abstract (ordinary differential) system in $Z(\tau)$

$$(3.1) \qquad \dot{z}(t) = \alpha(q)z(t) + \beta(q)u(t),$$

$$(3.2) \qquad z(0) = (\eta,\varphi),$$

$$(3.3) \qquad y(t) = C(q)z(t) + D(\alpha)\, u(t) \quad .$$

A mild solution to (3.1)-(3.2) is given by the variation of parameters formula

$$(3.4) \qquad z(t;\gamma,u) = S(t;q)(\eta,\varphi) + \int_0^t S(t-s;q)\beta(q)u(s)\, ds.$$

The following result is fundamental for all the approximation methods we consider. Its proof may be found in [4].

THEOREM 3.1. Suppose that $\gamma = (\eta,\varphi,q) \in \Gamma$ and $u \in L_2(0,T;R^m)$. If $x(t;\gamma,u)$ denotes the solution to the hereditary equation (2.1)-(2.2), then z defined by (3.4) satisfies

$$(3.5) \qquad z(t;\gamma,u) = (x(t;\gamma,u),x_t(\cdot;\gamma,u))$$

for all $t \geq 0$. In particular, the output to the abstract system (3.1)-(3.3) is the same as the output to the hereditary system (2.1)-(2.3).

It is clear from the above equivalence that the hereditary system (2.1)-(2.3) can be approximated by approximating the abstract system (3.1)-(3.3). In order to approximate (3.1)-(3.3), it follows from the formula (3.4) and the equivalence (3.5) that one must approximate the following;

    i) the initial data (η, φ),

    ii) the semigroup $\{S(t;q)\}_{t \geq 0}$,

    iii) the operators $B(q)$, $C(q)$.

The approximation of the initial data is accomplished by projecting (η, φ) onto a finite dimensional subspace of $Z(\tau)$. In order to approximate $S(t;q)$, recall that $S(t;q)$ is an evolution operator which is sometimes written

$$S(t;q) = e^{\alpha(q)t},$$

even though $\alpha(q)$ is unbounded. However, this (formal) identification illustrates the basic idea; $S(t,q)$ is approximated by approximating $\alpha(q)$. Similarly, we must approximate $B(q)$ and $C(q)$. Consequently, we construct approximating systems to (3.1)-(3.3) (and hence (2.1)-(2.3)) by

    1) projecting (η, φ) onto some finite dimensional sub-
        space of $Z(\tau)$,

and

    2) approximating the operators $\alpha(q)$, $B(q)$ and $C(q)$.

Although the above remarks are based on formal ideas, the

steps outlined above can be effected in a rigorous mathematical framework. We shall not attempt to develop the relevant theory in this paper; rather we refer the interested reader to [5].

We turn now to the two particular schemes that we have tested rather extensively on a number of numerical examples. The first method (AVE) is based on step-function approximations to the initial function φ, while the second method (SPLINE) is based on linear spline approximations to φ.

<u>AVE</u>: Corresponding to the partition $t_j^N = \frac{-j\tau}{N}$, $j = 0,1,\cdots,N$, of $[-\tau,0]$, we define the subspace $Z_A^N(\tau) = \{(\eta,\varphi) \in Z(\tau) \mid \varphi$ is a constant on each of the subintervals $[t_j^N, t_{j-1}^N)\}$. Let $P_A^N(\tau)$ be the orthogonal projection of $Z(\tau)$ onto the closed subspace $Z_A^N(\tau)$. In particular,

$$P_A^N(\tau)(\eta,\varphi) = (\eta,\varphi^N)$$

with

$$\varphi^N(s) = \sum_{j=1}^{N} \varphi_j^N \chi_j^N(s) \ ,$$

where $\chi_j^N(s)$ is the characteristic function for $[t_j^N, t_{j-1}^N)$ and φ_j^N is the mean (average) value

$$\varphi_j^N = \frac{N}{\tau} \int_{t_j^N}^{t_{j-1}^N} \varphi(s) \ ds \ ,$$

$j = 1, 2, \cdots N$. We take (η,φ^N) for our approximation to the initial data (η,φ). To approximate $Q(q)$, observe that the first

coordinate of $\alpha(q)(\eta,\varphi)$ is simply $L(q)\varphi$ and the second coordinate is $\dot{\varphi}$. Let $K_j^N(\alpha)$ be defined by

$$K_j^N(\alpha) = \int_{t_j^N}^{t_{j-1}^N} K(\alpha,s)\, ds = \int_{-r}^0 K(\alpha,s)\, \chi_j^N(s)\, ds$$

and suppose that $(v_0^N,\psi) = (v_0^N, \sum_{j=1}^N v_j^N \chi_j^N)$ belongs to $Z_A^N(\tau)$. Define $L_N(q)$ and $D_N(q)$ by

$$L_N(q)(v_0^N,\psi) = A_0(\alpha)\, v_0^N + A_1(\alpha)v_N^N + \sum_{j=1}^N K_j^N(\alpha) v_j^N$$

and

$$D_N(q)(v_0^N,\psi) = \sum_{j=1}^N \frac{N}{\tau}[v_{j-1}^N - v_j^N]\, \chi_j^N .$$

The operator $\alpha(q)$ is approximated by the finite dimensional operator $\alpha^N(q)$ on $Z_A^N(\tau)$ defined by

$$\alpha^N(q)(v_0^N,\psi) = (L_N(q)(v_0^N,\psi),\ D_N(q)(v_0^N,\psi)) .$$

If an appropriate basis for $Z_A^N(\tau)$ is selected, then $R^{(N+1)n}$ can be identified with $Z_A^N(\tau)$ and the above scheme leads to the following approximating system. For $N \geq 1$ define the $[(N+1)n] \times [(N+1)n]$ matrix $A^N(q)$ by

$$(3.6) \quad A^N(q) = \begin{bmatrix} A_0(\alpha) & K_1^N(\alpha) & \cdot & \cdot & \cdot & K_{N-1}^N(\alpha) & A_1(\alpha)+K_N^N(\alpha) \\ \frac{N}{\tau}I & -\frac{N}{\tau}I & 0 & & \cdot & \cdot & \cdot & 0 \\ 0 & \frac{N}{\tau}I & -\frac{N}{\tau}I & & \cdot & \cdot & \cdot \\ \cdot & & \cdot & \cdot & & & \cdot \\ \cdot & & & \cdot & \cdot & & \cdot \\ \cdot & & & & \cdot & \cdot \\ 0 & \cdot & \cdot & & \cdot & 0 & \frac{N}{\tau}I & -\frac{N}{\tau}I \end{bmatrix},$$

where I is the n x n identity matrix. The $[(N+1)n] \times m$ matrix $B^N(q)$ and the $k \times [(N+1)n]$ matrix $C^N(q)$ are defined by

$$(3.7) \quad B^N(q) = \begin{bmatrix} B(\alpha) \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \quad \text{and} \quad C^N(q) = [C(\alpha) \ 0 \ \dots \ 0] ,$$

respectively.

The approximating ordinary differential system (of dimension $(N+1)n$ is given by

$$(3.8) \quad \dot{z}^N(t) = A^N(q)z^N(t) + B^N(q)u(t)$$

$$(3.9) \quad z^N(0) = z_0^N$$

$$(3.10) \quad y^N(t) = C^N(q)z^N(t) + D(\alpha)u(t) ,$$

where $z_0^N = \mathrm{col}(\eta, \varphi_1^N, \cdots, \varphi_N^N)$. The approximate system (3.8) - (3.10) will be referred to as the AVE approximation to (2.1) - (2.3).

SPLINE: Corresponding to the partition $t_j^N = \frac{-j\tau}{N}$, $j = 0, 1, \cdots, N$, of $[-\tau, 0]$ we define the subspace $Z_S^N(\tau) = \{(\varphi(0), \varphi) \in Z(\tau) | \varphi$ is a first order (piecewise linear) spline function with knots at $t_j^N, i = 0, 1, \ldots, N.\}$. Let $P_S^N(\tau)$ be the orthogonal projection of $Z(\tau)$ onto $Z_S^N(\tau)$. Note that if $(\eta, \varphi) \in Z(\tau)$, then $P_S^N(\tau)(\eta, \varphi)$ belongs to $\mathscr{D}(\mathcal{Q}(q))$. One can argue that $P_S^N(\tau)(\eta, \varphi) \to (\eta, \varphi)$ as $N \to +\infty$ and hence it is not unreasonable to expect that for $(\eta, \varphi) \in \mathscr{D}(\mathcal{Q}(q))$

$$\lim_{N \to +\infty} P_S^N(\tau) \, \mathcal{Q}(q) P_S^N(\tau)(\eta, \varphi) = \mathcal{Q}(q)(\eta, \varphi) .$$

Consequently, we define $\mathcal{Q}^N(q) : Z_S^N(\tau) \to Z_S^N(\tau)$ by

$$(3.11) \qquad \mathcal{Q}^N(q) = P_S^N(\tau) \, \mathcal{Q}(q) \, P_S^N(\tau) .$$

In order to represent the operator $\mathcal{Q}^N(q)$ and construct an ordinary differential system in Euclidean space, we follow the general outline given by Banks and Kappel [8]. Let $e_j^N, j = 0, 1, \cdots, N$ denote the scalar first order spline function on $[-\tau, 0]$ characterized by

$$e_j^N(t_i^N) = \delta_{ij}, \quad i, j = 0, 1, \cdots N ,$$

where δ_{ij} is the Kronecker symbol. The matrix $\beta^N = [\beta_1^N, \cdots, \beta_{N+1}^N]$ defined by

$$\beta^N = [e_0^N, e_1^N, \cdots, e_N^N] \otimes I$$

(where $\otimes$ denotes the Kronecker product) is such that the set

$$\hat{\beta}^N_j = (\beta^N_j(0), \beta^N_j)$$

$j = 1, 2, \ldots, N+1$, forms a basis for $Z^N_S(\tau)$. With this basis $Z^N_S(\tau)$ is identified with $R^{(N+1)n}$ and the following system may be constructed.

Let $Q^N(q)$ be the $[(N+1)n] \times [(N+1)n]$ matrix

$$(3.12) \quad Q^N(q) = \frac{\tau}{N} \begin{bmatrix} \frac{N}{\tau}+\frac{1}{3} & \frac{1}{6} & 0 & \cdot & \cdot & \cdot & 0 \\ & & & & & & \cdot \\ & & & & & & \cdot \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & & & & \cdot \\ & \cdot & & \cdot & & & 0 \\ & & \cdot & & \cdot & & \\ & & & \cdot & & \cdot & \\ 0 & & \frac{1}{6} & & \frac{2}{3} & & \frac{1}{6} \\ \cdot & & & & & & \\ \cdot & & & & & & \\ 0 & \cdot & \cdot & \cdot\cdot & 0 & \frac{1}{6} & \frac{1}{3} \end{bmatrix} \otimes 1$$

and

$$(3.13) \qquad\qquad H^N(q) = H^N_1(q) + H^N_2(q)$$

where

$H^N_1(q)$ and $H^N_2(q)$ are $[(N+1)n] \times [(N+1)n]$ matrices defined by

$$(3.14) \quad H_1^N(q) = \begin{bmatrix} A_0(\alpha) + \widetilde{K}_0^N(\alpha) & \widetilde{K}_1^N(\alpha) & \cdots & \widetilde{K}_{N-1}^N(\alpha) & A_1(\alpha) + \widetilde{K}_N^N(\alpha) \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 \end{bmatrix}$$

with

$$\widetilde{K}_j^N(\alpha) = \int_{-r}^{0} K(\alpha,s) \, e_j^N(s) \, ds \, ,$$

and

$$(3.15) \quad H_2^N(q) = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & & & 0 \\ \frac{1}{2} & 0 & -\frac{1}{2} & & & 0 \\ & & & & & \\ 0 & & & \frac{1}{2} & 0 & -\frac{1}{2} \\ & & & & & \\ 0 & & & 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \otimes I$$

It follows (see [8]) that $\alpha^N(q) = P_S^N(\tau)\alpha(q)P_S^N(\tau)$ has the $[(N+1)n] \times [(N+1)n]$ matrix representation

$$(3.16) \qquad \widetilde{A}^N(q) = [Q^N(q)]^{-1} H^N(q) \ .$$

Define the $[(N+1)n] \times m$ matrix $\widetilde{B}^N(q)$ and the $k \times [(N+1)n]$ matrix $\widetilde{C}^N(q)$ by

$$(3.17) \qquad \widetilde{B}^N(q) = [Q^N(q)]^{-1} \begin{bmatrix} B(\alpha) \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix} \quad \text{and} \quad \widetilde{C}^N(q) = [C(\alpha) \ 0 \cdots 0],$$

respectively. Any vector $z^N \in Z_S^N(\tau)$ can be written as

$$z^N = \sum_{j=0}^{N} (e_j^N(0), e_j^N) \xi_j^N \ ,$$

where $\xi_j^N \in R^n$. Thus, the vector $z^N \in Z_S^N(\tau)$ can be identified with the vector $\mathrm{col}(\xi_0^N, \xi_1^N, \cdots \xi_N^N)$ in $R^{(N+1)n}$.

The approximating ordinary differential system becomes

$$(3.18) \qquad \dot{z}^N(t) = \widetilde{A}^N(q) z^N(t) + \widetilde{B}^N(q) u(t) \ ,$$

$$(3.19) \qquad z^N(0) = z_0^N \ ,$$

$$(3.20) \qquad y^N(t) = \widetilde{C}^N(q) w^N(t) + D(\alpha) u(t) \ ,$$

where z_0^N is the vector in $R^{(N+1)n}$ identified with $P_S^N(\tau)(\eta, \varphi) \in Z_S^N(\tau)$. We shall refer to the system $(3.18) - (3.20)$ as the SPLINE approximation scheme for $(2.1) - (2.3)$.

REMARK. When making computations involving the spline system $(3.18)-(3.20)$ one never actually computes $[Q^N(q)]^{-1}$ but rather

solves

$$[Q^N(q)]w^N = H^N(q)v^N$$

directly in order to obtain $w^N = \widetilde{A}^N(q)v^N$.

Both of the above schemes (AVE and SPLINE) fall within the general theoretical framework presented in [5], where convergence results and error estimates are given.

4. The approximating problems.

The system (3.8)-(3.10) will be called AVE and the system (3.18)-(3.20) will be called SPLINE. Both are ordinary differential systems (of dimension (N+1)n) that approximate the dynamical response of the hereditary system (2.1)-(2.3). In order to state an approximating identification problem, we must approximate the constraint set Γ.

Suppose that $\gamma = (\eta,\varphi,q) \in \Gamma$ and that $P_A^N(\tau)(\eta,\varphi)$ and $P_S^N(\tau)(\eta,\varphi)$ have the representations in $R^{(N+1)n}$ given by

$$P_A^N(\tau)(\eta,\varphi) = (\tau,\varphi_1^N,\cdots,\varphi_N^N)$$

$$P_S^N(\tau)(\eta,\varphi) \cong (\xi_0^N,\xi_1^N,\cdots,\xi_N^N) \ .$$

Let Π_A^N and Π_S^N denote the mappings from $Z \times R^\mu \times R$ to $R^{(N+1)n} \times R^\mu \times R$ defined by

$$\Pi_A^N(\eta,\varphi,\alpha,\tau) = (P_A^N(\tau)(\eta,\varphi),\alpha,\tau)$$

and

$$\pi_S^N(\eta,\varphi,\alpha,\tau) = (P_S^N(\tau)(\eta,\varphi),\alpha,\tau) .$$

Thus motivated, we choose the approximating constraint sets as $\bar{\Gamma}_A^N = \bar{\pi}_A^N \Gamma$ and $\bar{\Gamma}_S^N = \bar{\pi}_S^N \Gamma$. For each of the systems (3.8) - (3.10) and (3.18) - (3.20) we have the following approximating parameter identification problem:

PROBLEM (IDN). <u>Given</u> <u>the</u> <u>control</u> u <u>in</u> $L_2(0,T;R^m)$ <u>and</u> <u>observations</u> $\bar{y}_i \in R^k$ <u>at</u> <u>times</u> t_i, <u>find</u> <u>the</u> <u>parameters</u> $\hat{\gamma}_A^N \in \Gamma_A^N$ $(\hat{\gamma}_S^N \in \Gamma_S^N)$ <u>which</u> <u>minimizes</u> <u>the</u> <u>fit</u> <u>error</u>

$$(4.1) \qquad E^N(\gamma^N) = \frac{1}{2} \sum_{i=1}^M | y^N(t_i;\gamma^N,u) - \bar{y}_i |^2$$

<u>where</u> $y^N(t;\gamma^N,u)$ <u>is</u> <u>the</u> <u>output</u> <u>to</u> <u>the</u> <u>AVE system</u> (3.8)-(3.10) (<u>SPLINE system</u> (3.18)-(3.20)), <u>and</u> <u>the</u> <u>minimization</u> <u>takes</u> <u>place</u> <u>over</u> Γ_A^N (<u>over</u> Γ_S^N).

Given a parameter $\gamma = (\eta,\varphi,\alpha,\tau)$, one may construct the approximating systems AVE and SPLINE, corresponding to γ. Let G^N and W^N be the (N+1)n square matrices defined by

$$
G^N = \begin{bmatrix}
G & 0 & . & . & . & 0 \\
0 & 0 & . & . & . & . \\
. & . & . & . & . & . \\
. & . & . & . & . & . \\
. & . & & & & . \\
0 & 0 & . & . & . & 0
\end{bmatrix}
$$

and

$$
W^N = \begin{bmatrix}
W & 0 & . & . & . & 0 \\
0 & 0 & . & . & & . \\
. & . & . & . & & . \\
. & . & & . & & . \\
. & . & & & . & . \\
0 & 0 & . & . & . & 0
\end{bmatrix} .
$$

For each of the approximating systems AVE and SPLINE we have the following approximating optimal control problems:

PROBLEM (OCN). Given the system parameters $\gamma \in \Gamma$, find a control $\hat{u}_A^N$ ($\hat{u}_S^N$) in $L_2(0,T;R^m)$ which minimizes the performance criterion

$$
(4.2) \quad J^N(u) = \tfrac{1}{2}[(z^N(T))^T G^N z^N(T)] + \tfrac{1}{2} \int_0^T \{(z^N(s))^T W^N z^N(s) + u^T(s) R u(s)\} ds,
$$

where $z^N(t) = z^N(t;\gamma,u)$ is the solution to the AVE system (3.8)-(3.10) (SPLINE system (3.18)-(3.20)). The optimal cost will be denoted by $\mathfrak{J}^N$ (i.e. $\mathfrak{J}^N = J^N(\hat{u}^N)$).

For each N, the approximating identification and optimal control problems are now finite dimensional in the sense that the dynamical systems AVE and SPLINE are governed by ordinary differential equations in $R^{(N+1)n}$. The basic idea is to solve (for fixed N) each of these problems to obtain $\hat{\gamma}_A^N$, $\hat{\gamma}_S^N$, $\hat{u}_A^N$ and $\hat{u}_S^N$. It can be shown that under reasonable conditions $\hat{\gamma}_A^N$ and $\hat{\gamma}_S^N$ "converge" to $\gamma*$ and $\hat{u}_A^N$ and $\hat{u}_S^N$ converge to u* (see [5] for a precise statement of the results).

The remainder of this paper is devoted to the study of numerical examples. In particular, we compute $\hat{\gamma}_A^N$, $\hat{\gamma}_S^N$, $\hat{u}_A^N$ and $\hat{u}_S^N$ and compare these values with the optimal values $\gamma*$ and u* for a number of hereditary systems. In doing so, we hope to demonstrate that the method is feasible to implement and that acceptable convergence rates are obtained.

All of the numerical results presented in the next sections were produced by computer programs written at Virginia Tech and Brown University. The identification problems were run at Virginia Tech on an IBM 370/158 computer. A maximum likelihood (least squares) algorithm was used to solve the approximating problem (IDN). A complete discription of the method and listing of the code may be found in the report [9]. The optimal control problems were run at Brown University on an IBM 360/67 computer. For both the linear and nonlinear control examples a conjugate-gradient minimization algorithm (as described in [1]) was used to solve the approximating problem (OCN).

5. Numerical solutions to the identification problem.

In this section we present a number of numerical results for the identification problem (PROBLEM (ID)), that are based on the approximation schemes (AVE and SPLINE) outlined in the previous sections. In order to generate most of the data for testing the algorithm we select a "true" set of parameters $\gamma^* = (\eta^*, \varphi^*, \alpha^*, \tau^*)$ and a control u and use the method of steps [11] to solve for x on the interval $[0,T]$.

In all of the examples presented below we used $\tau^* = 1$ and $u = u_\ell$, where u_ℓ is the unit step at $t = \ell$ defined by

$$u_\ell(t) = \begin{cases} 0 & t < \ell, \\ \\ 1 & \ell \le t, \end{cases}$$

and $0 < \ell < 1$. The final time of $T = 2$ was used in most of the examples (except 05.1). The observations $\bar{y}_i = y(t_i)$ were generated at 101 equally spaced time steps on $[0,T]$. In some examples noise was added to the model to produce "noisy observations"

$$\bar{y}(t) = y(t) + \nu(t),$$

where $\nu(t) = \text{col}(\nu_1(t), \cdots, \nu_k(t))$ is a computer simulated vector of normal random variables $\nu_i(t)$ (routine GGNQF of the IMSL library, see IMSL Users Guide), each with zero mean and preset standard variation.

For each fixed N, the approximation Problem (IDN) was solved using a maximum likelihood estimator (MLE). Since the MLE is an iterative procedure it is necessary to supply a startup (i.e. an initial guess) for the optimal parameter $\hat{\gamma}_A^N$ (or $\hat{\gamma}_S^N$ for the spline scheme). If β denotes an unknown parameter to be estimated (i.e. $\beta = \alpha$ or $\beta = \tau$, etc), then $\beta^{N,I}$ will denote the estimate for $\hat{\beta}^N$ obtained after I iterations of the MLE applied to PROBLEM (IDN). The startup value will be denoted by $\beta^{N,0}$.

It is helpful to understand the numbering system for the identification examples. The first two characters in the example number indicate what model is used for the generation of data. The number after the decimal point refers to the specific numerical run. For example, all "S2" examples are problems where the "true" system is governed by

$$\dot{x}(t) = .05\ x(t) - 4.0\ x(t-1) + u_{.1}(t)\ ,$$

$$x_0(s) \equiv 1\ ,\ -1 \leq s \leq 0\ ,$$

$$y(t) = x(t)\ .$$

In EXAMPLE S2.1 we assume that $a_1^* = -4.0$ is unknown and attempt to estimate this parameter, while in EXAMPLE S2.2 we assume that the time delay $r^* = 1.0$ is unknown and estimate this parameter, etc.

The figures are labeled at the top. In the left hand corner
the lettering indicates the example number, the value of N and the
approximation scheme. For example, S2.1N16A refers to Example
S2.1, N = 16 and the AVE procedure. The lettering in the right
hand corner represents iteration number in the MLE algorithm.

ID MODEL S2

This is a simple scalar model with discrete delay. The system is governed by

$$\dot{x}(t) = .05\ x(t) - 4.0\ x(t-1) + u_{.1}(t),$$

with initial data

$$x(0) = 1 \quad \text{and} \quad x_0(s) \equiv 1, \quad -1 \leq s < 0.$$

The output is simply the state at time t, viz:

$$y(t) = x(t).$$

As described above, this system was analytically integrated (using the method of steps) to construct the solution on $[0,2]$. The resulting solution was evaluated at 101 equally spaced points to generate data for the following four examples; S2.1 - S2.4.

EXAMPLE S2.1

In this initial example we consider the problem of identifying
a single parameter, namely the coefficient of the delayed term in
the model S2. Thus, in our parametric model the dynamics are des-
cribed by

$$\dot{x}(t) = .05\ x(t) + a_1 x(t-1) + u_{.1}(t)$$

with initial condition

$$x(0) = 1\ ,\ x_0(s) = 1\ ,\ -1 \leq s < 0\ ,$$

and output

$$y(t) = x(t)\ .$$

For N = 2, 4, 8, 16 and 32 the resulting Problems (IDN) were
"solved" using a version of the computer code described in [9].
Since the numerical algorithm is iterative, it is necessary to
provide a "start-up" value for a_1 and in this example we used

$$a_1^{N,0} = 0.0\ .$$

The algorithm provides a sequence of "improved" estimates for $\hat{a}_1^N$
and will terminate when either a maximum number of iterations is
achieved, or when the norm of the gradient of E^N is less than 10^{-3}.
In the latter case we claim that the procedure has "converged".

Results of the numerical experiments for this example are shown in Table S2.1.1. Notice that for N = 2, the AVE procedure had not "converged" at 10 iterations, and that the SPLINE estimate of the parameter at N = 4 is better than the AVE estimate at N = 32.

By using the parameter values we can estimate the rate of convergence as

$$\delta = \frac{\ln \left[\|e_N\| / \|e_{2N}\| \right]}{\ln 2}$$

where $e_N = \hat{\gamma}^N - \gamma*$ is the error. From N = 2 in the SPLINE estimates we find $\delta \cong 1.7$, while for N = 4 in the AVE result we estimate $\delta \cong .12$. Such estimates for the rates of convergence must be viewed with caution because the numerical values are corrupted by sources of error other than the approximation scheme.

Figures S2.1.1 and S2.1.2 show the converged data fits at N = 16 for AVE and SPLINE, respectively. The N = 32 results are essentially the same as those for N = 16.

Since computer requirements are of practical interest, we note that for N = 32, the AVE algorithm took about 15 sec. per iteration, while the SPLINE procedure required a little over 16 sec. per iteration. This comparison is not completely fair because the majority of the code used is common to both AVE and SPLINE and it is structured to provide the generality needed for

the SPLINE method. A streamlined code for AVE alone might pro-
duce as much as a 20% savings in execution time.

AVE			SPLINE		
$\underline{N}$	$\hat{a}_1^N$	$\vert e_N \vert$	$\underline{N}$	$\hat{a}_1^N$	$\vert e_N \vert$
2	did not converge		2	-4.1655	.1655
4	-4.1144	.1144	4	-4.0505	.0505
8	-4.1050	.1050	8	-4.0208	.0208
16	-4.0852	.0852	16	-4.0139	.0139
32	-4.0584	.0584	32	-4.0122	.0122
$\alpha_1^* =$	-4.0000		$\alpha_1^* =$	-4.0000	

TABLE S2.1.1
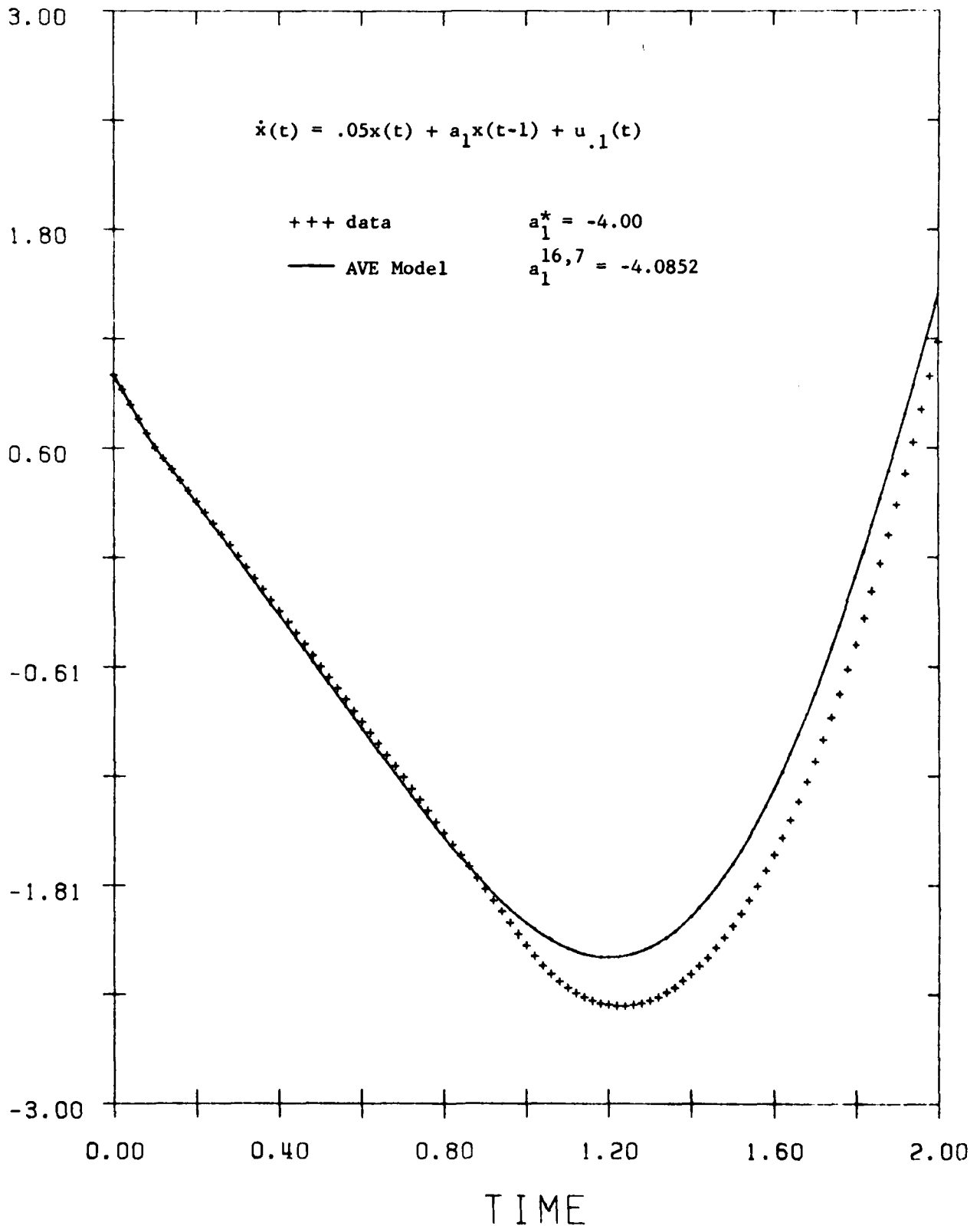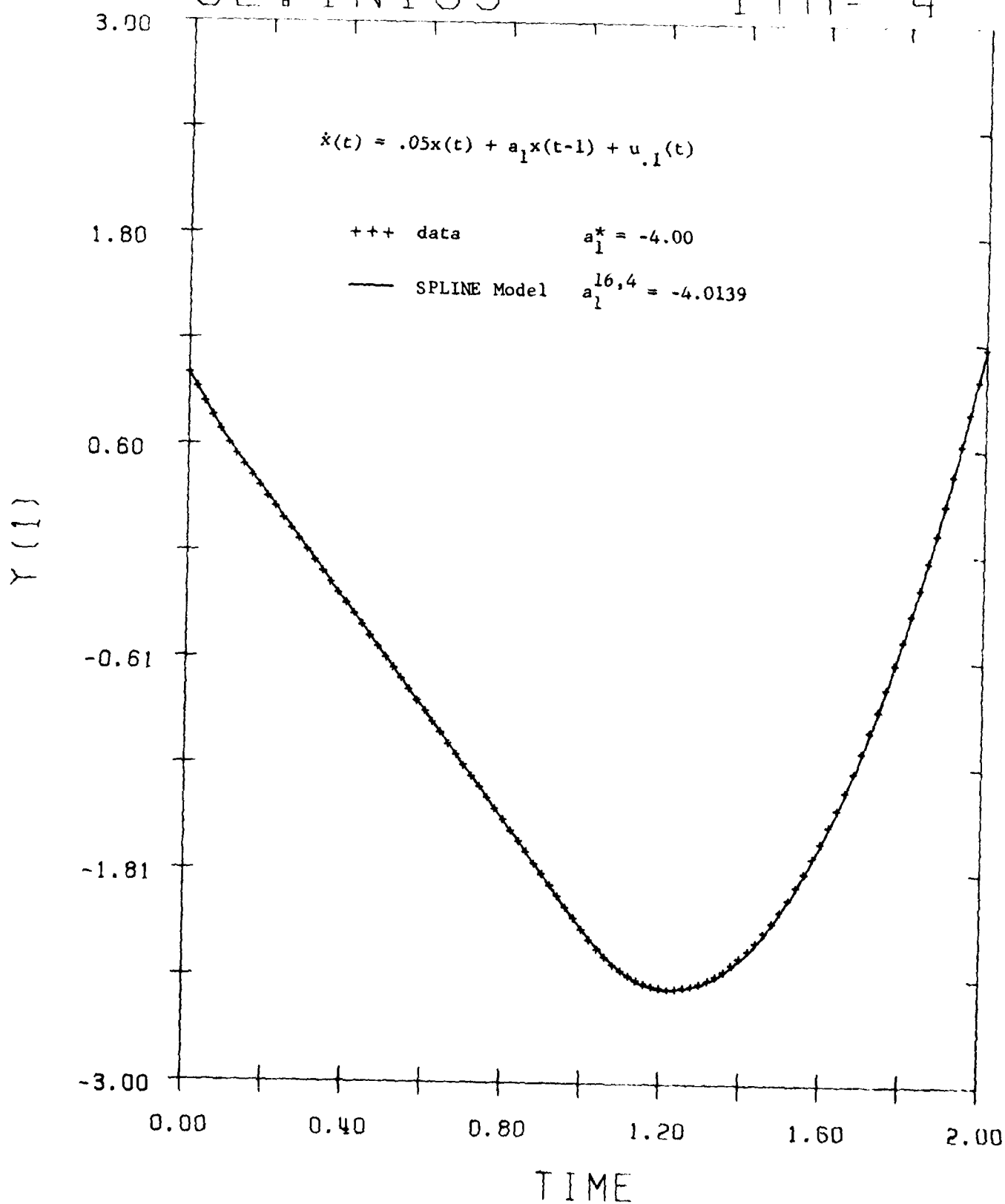
FIGURE S2.1.1

FIGURE S2.1.2

EXAMPLE S2.2

In this example we consider the problem of identifying the time delay alone, with all other parameters known. Thus, the parametric model of the system is

$$\dot{x}(t) = .05 \, x(t) - 4.0 \, x(t-r) + u_{.1}(t)$$

with initial condition

$$x(0) = 1 , \quad x_0(s) \equiv 1 , -r \leq s < 0 ,$$

and output

$$y(t) = x(t) .$$

As before, we conducted numerical experiments for $N = 2, 4, 8,$ 16 and 32. Our start-up was

$$r^{N,0} = .5 ,$$

while the true value is of course $r* = 1.0$. At $N = 2$ and $N = 4$ an interesting phenomenon appeared; namely, for the start up value of $r^{N,0} = 0.5$ the AVE procedure "converged" and the SPLINE procedure "diverged". To examine the causes of this result we evaluated the cost function E^4 for AVE and SPLINE at a variety of r values. The interesting results of this investigation are shown in Figure S2.2.1. It happens that the SPLINE cost function is more "oscillatory" than the AVE cost function for $N = 4$. Both

have two local minima. However, the SPLINE cost function is such that the start up $r^{4,0} = 0.5$ is not in the valley of the "global" minimum $(\hat{r}_S^4 = .9972)$, while for the AVE procedure, $r^{4,0} = 0.5$ is in the valley of the global minimum $(\hat{r}_A^4 = 1.2001)$.

When the cost function has more than a single local minimum, the system is said to suffer a lack of (global) identifiability (at least for the specified input). In such cases it is important to have good start-up values for the parameters.

Table S2.2.1 illustrates the convergence for this example. Note that for the reasons outlined above, different start-up values were used for AVE and SPLINE. Again the results show that for N = 2 the SPLINE algorthim gives better estimates of the parameter r than AVE for N = 32. Figures S2.2.2 and S2.2.3 show the N = 4 data fits for AVE and SPLINE, respectively.

	AVE	$r^{N,0} = 0.5$		SPLINE	$r^{N,0} = 0.8$
N	$\hat{r}^N$	$|e_N|$	N	$\hat{r}^N$	$|e_N|$
2	1.4603	.4603	2	1.0084	.0084
4	1.2001	.2001	4	.9972	.0028
8	1.0923	.0923	8	.9983	.0017
16	1.0439	.0439	16	.9986	.0014
32	1.0212	.0212	32	1.0018	.0018
r* =	1.0000		r* =	1.0000	

TABLE S2.2.1

COST N=4
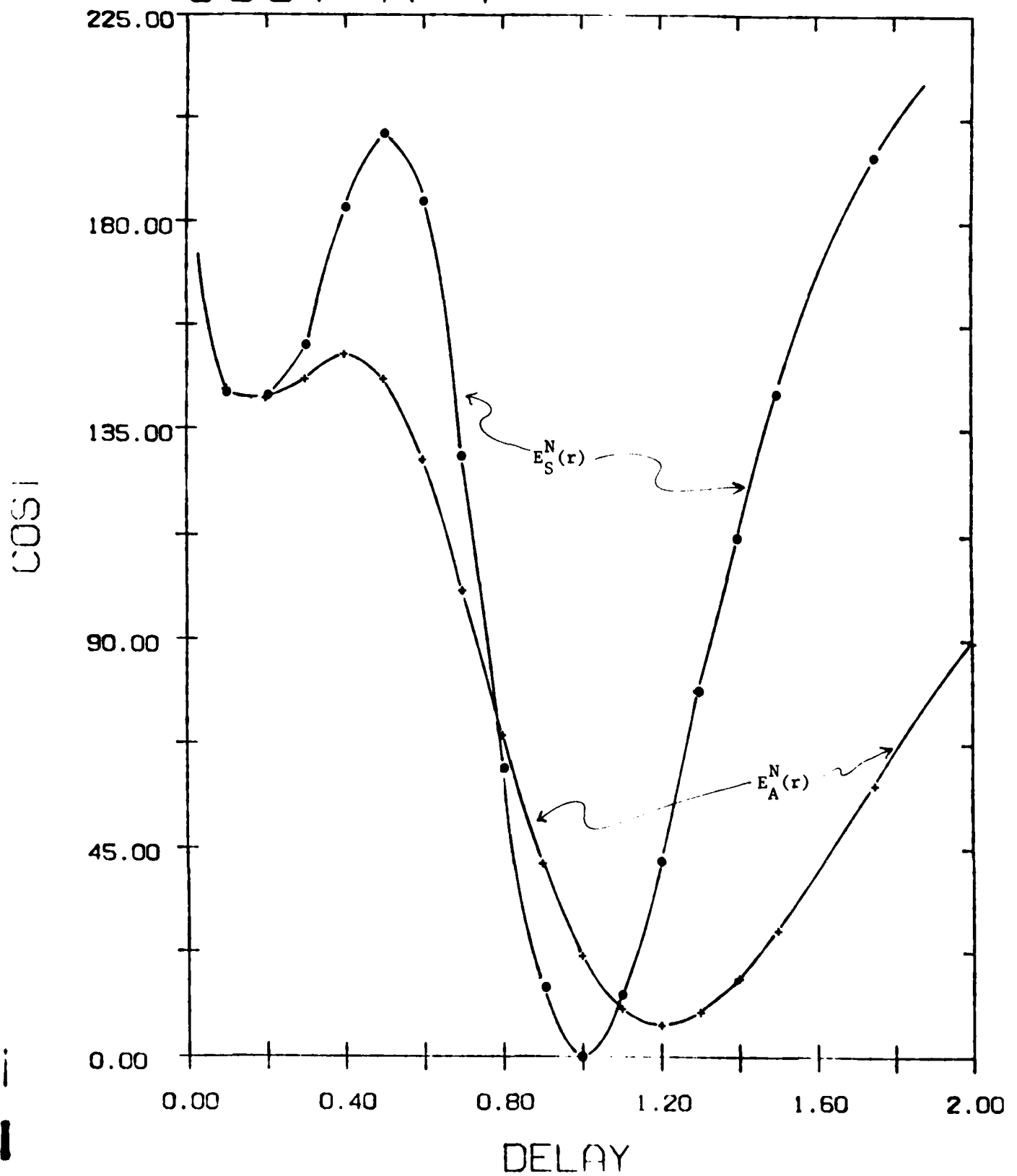
FIGURE S2.2.1

S2.2N4AV   ITR= 6

$$\dot{x}(t) = .05x(t) - 4x(t-r) + u_{.1}(t)$$

+++   data   r* = 1.0

——   AVE Model $r^{4,6} = 1.2001$
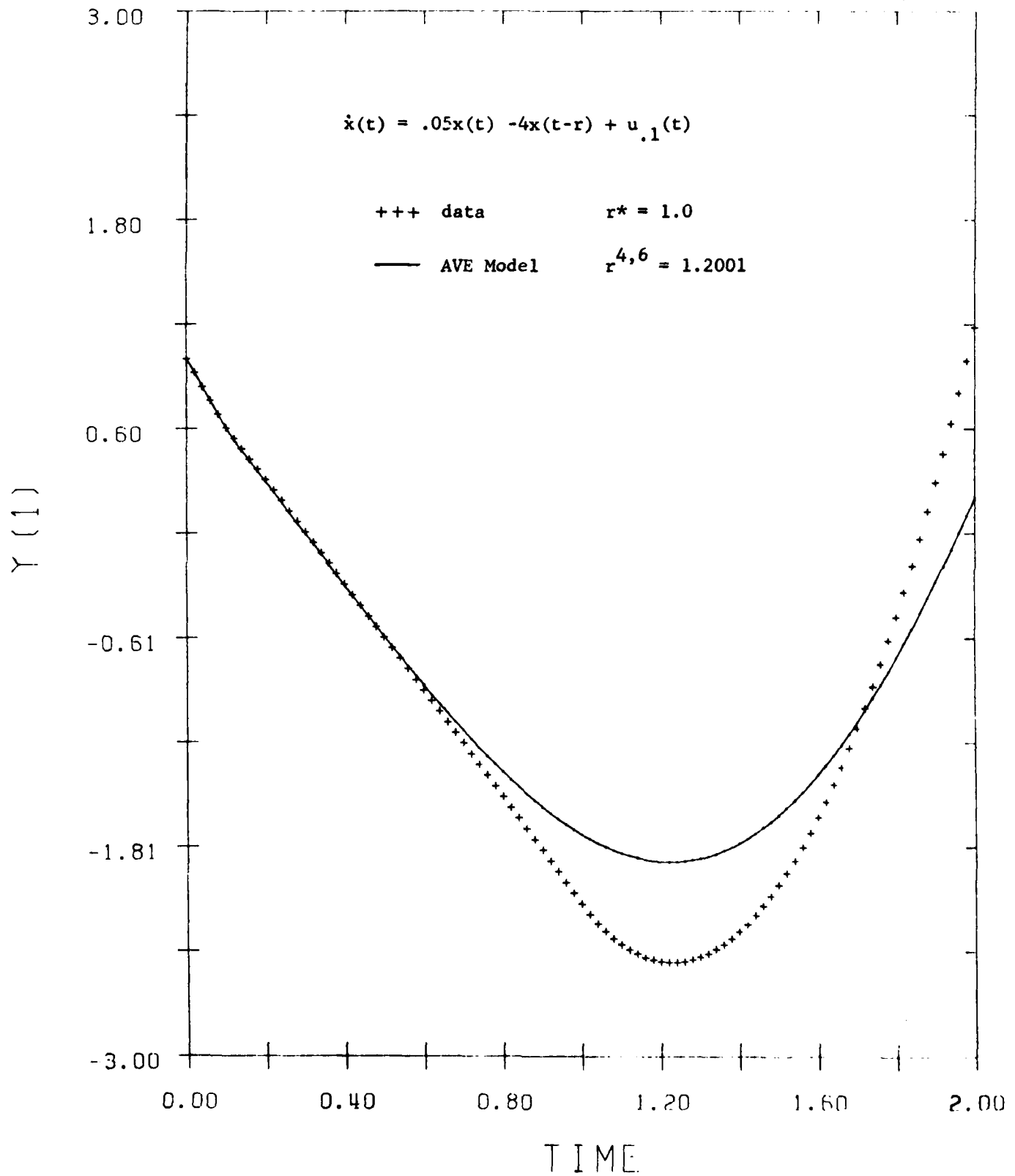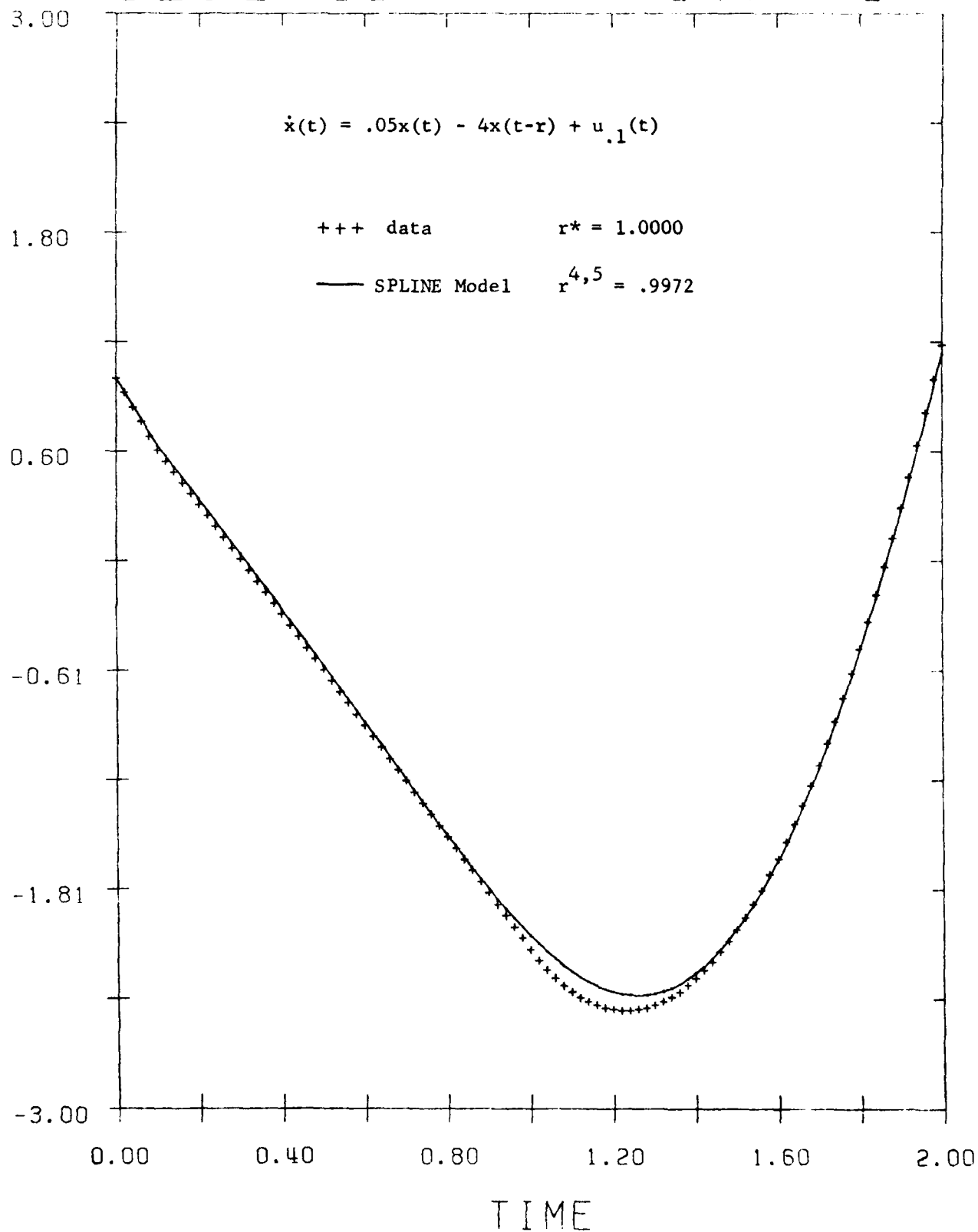
FIGURE S2.2.2

FIGURE S2.2.3

EXAMPLE S2.3

In this example we consider the problem of estimating the two coefficients in the model. Therefore, the system is modeled by

$$\dot{x}(t) = a_0 x(t) + a_1 x(t-1) + u_{.1}(t)$$

with initial data

$$x(0) = 1 , \quad x_0(s) \equiv 1 , -1 \leq s < 0 ,$$

and output

$$y(t) = x(t) .$$

Numerical runs for N = 2, 4, 8, 16 and 32 were conducted. The start-up values for $a_0^* = .05$ and $a_1^* = -4.0$ were chosen to be

$$a_0^{N,0} = .03 \quad \text{and} \quad a_1^{N,0} = -3.0 .$$

Table S2.3.1 contains a summary of the estimates for both AVE and SPLINE. The ℓ_1 errors $(|\hat{a}_0^N - a_0^*| + |\hat{a}_1^N - a_1^*|)$ are given in Table S2.3.2. Note that the SPLINE estimate at N = 4 is better than the AVE estimate at N = 32.

Figures S2.3.1 and S2.3.2 show the converged data fits at N = 16 for AVE and SPLINE, respectively. Observe that the SPLINE procedure provides almost a "perfect" match to the data.

AVE			SPLINE		
N	$\hat{a}_0^N$	$\hat{a}_1^N$	N	$\hat{a}_0^N$	$\hat{a}_1^N$
2	1.0869	-4.6236	2	.0995	-4.1639
4	.6525	-4.3160	4	.0417	-4.0523
8	.3825	-4.1660	8	.0439	-4.0222
16	.2245	-4.0898	16	.0449	-4.0151
32	.1384	-4.0505	32	.0454	-4.0133
$\gamma^* =$	.0500	-4.0000	$\gamma^* =$	.0500	-4.0000

TABLE S2.3.1

AVE		SPLINE	
N	$|e_N|$	N	$|e_N|$
2	1.6605	2	.2134
4	.9185	4	.0606
8	.4985	8	.0283
16	.2643	16	.0202
32	.1389	32	.0179

TABLE S2.3.2

S2.3N16A   ITR= 4

$$\dot{x}(t) = a_0 x(t) + a_1 x(t-1) + u_{.1}(t)$$

+ + + data $a_0^* = .05$ $a_1^* = -4.0$

——— AVE Model $a_0^{16,4} = .2245$ $a_1^{16,4} = -4.0898$
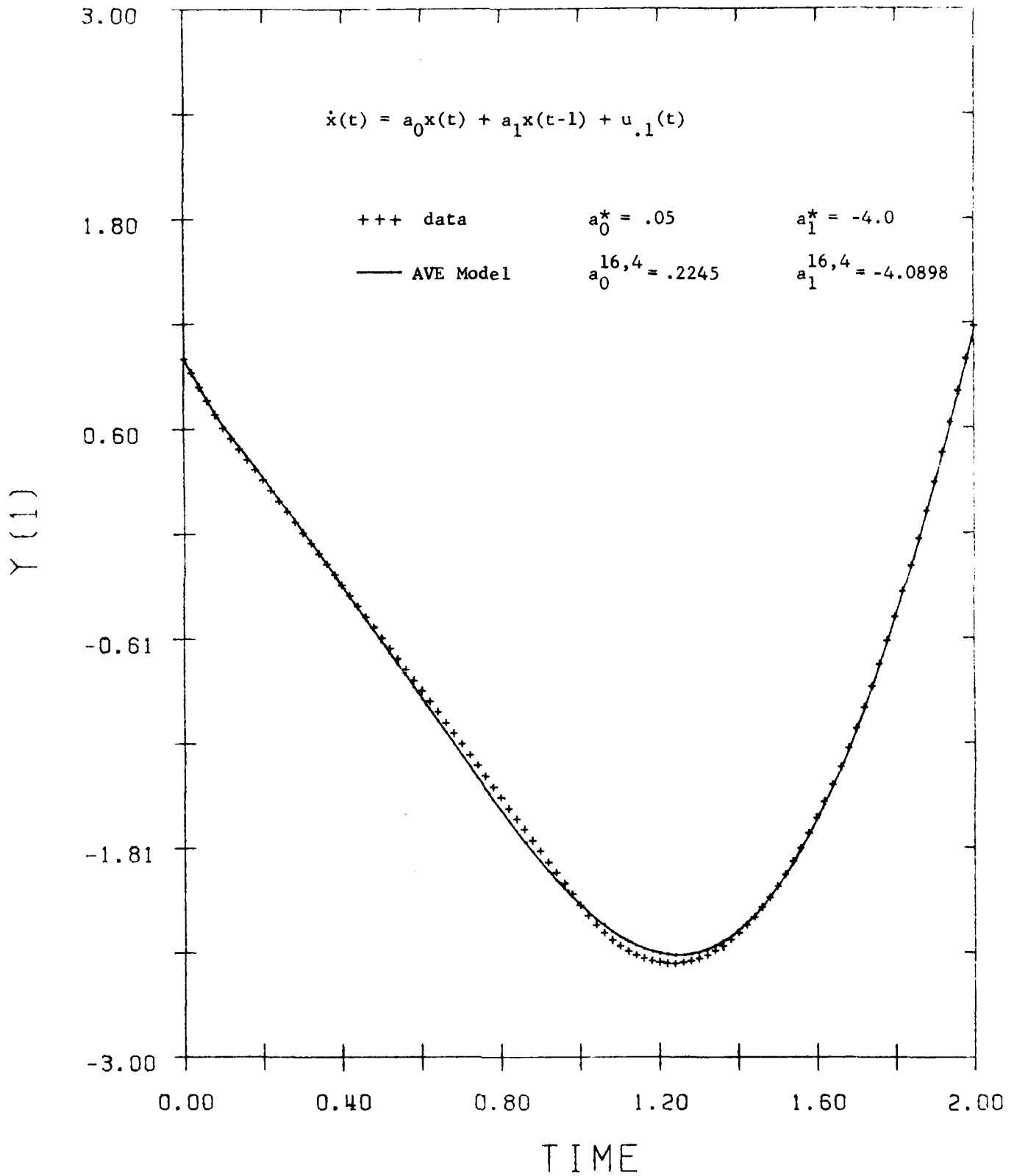
FIGURE S2.3.1

S2.3N16S   ITR= 4

$$\dot{x}(t) = a_0 x(t) + a_1 x(t-1) + u_{.1}(t)$$

+ + +   data $a_0^* = .05$ $a_1^* = -4.0$

——— SPLINE Model $a_0^{16,4} = .0449$ $a_1^{16,4} = -4.0151$

Y(1)

TIME
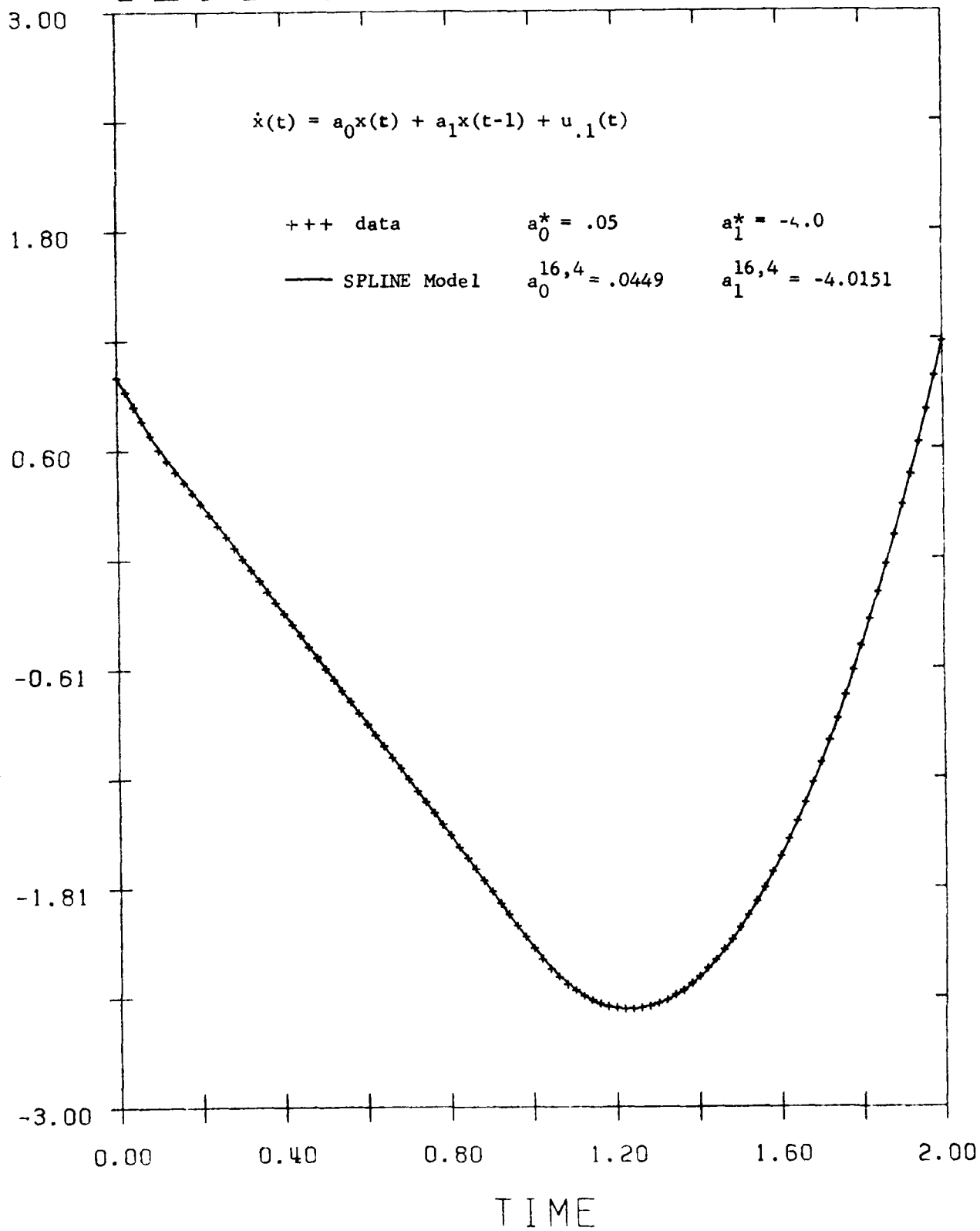
FIGURE S2.3.2

EXAMPLE S2.4

This is the first example in which the "original" identification problem is truly infinite dimensional. In particular, we seek to estimate the initial data $(\eta, \varphi) \in R \times L_2(-1, 0; R)$ and the coefficient of the delayed term. Thus our model is described by the scalar equation

$$\dot{x}(t) = .05\, x(t) + a_1 x(t-1) + u_{.1}(t),$$

with (unknown) initial data

$$x(0) = \eta, \quad x_0(s) = \varphi(s), \quad -1 \le s < 0,$$

and output

$$y(t) = x(t).$$

For each $N = 2, 4, 8, 16$ and 32, the approximating problem (IDN) was formulated as discussed in Section 4. Thus, for AVE we seek the "parameter"

$$\hat{\gamma}_A^N = (\eta, \varphi_1^N, \varphi_2^N, \cdots, \varphi_N^N, a_1),$$

where $(\eta, \varphi_1^N, \varphi_2^N, \cdots, \varphi_N^N)$ represents the projection of the initial data. Similarly, for SPLINE we seek the "parameter"

$$\hat{\gamma}_S^N = (\xi_0^N, \xi_1^N, \cdots, \xi_N^N, a_1),$$

where $(\xi_0^N, \xi_1^N, \cdots, \xi_N^N)$ represents the SPLINE projection of the initial data. The "start-up" for $(\eta, \varphi) \in R \times L_2(-1, 0; R)$ is the zero

initial data $(0,0)$, while its true value is $(1, 1)$. The "start-up" for $a_1^* = -4$ is

$$a_1^{N,0} = -3.0 \ .$$

Table S2.4.1 provides an overview of the results. Because the initial data is in $R \times L_2(-1, 0; R)$ we have only displayed the Z-norm of the error and the estimated value for a_1^N. The comparison of the two schemes is quite striking, particularly the relative ability to estimate the initial data. Shown in Figure S2.4.1 are graphs of the true initial data and the corresponding estimates produced by AVE and SPLINE for $N = 4$. It is apparent at least for the chosen "start-up" values that the SPLINE procedure readily finds good estimates for the parameters, while the AVE scheme has considerable difficulty.

It is interesting to compare the sequences of data fits generated as the iteration procedure evolves. Figures S2.4.2 through S2.4.4 show the data matches from the AVE algorithm for iterations 0, 4 and 9, respectively. From the match at iteration 4 (Figure S2.4.3) it might be deduced that AVE is in trouble. However, at iteration 9 the fit is quite good and Figure S2.4.4 does not give any hint of the poor values of the parameters indicated in Table S2.4.1.

Figures S2.4.5 through S2.4.7 illustrate the SPLINE data matches at iterations 0, 4, and 9 respectively. Again the iteration

4 matches indicate some difficulty while by iteration 9 the match
is quite good. It happens that the SPLINE estimates of the para-
meters are excellent.

Although one can not be certain, it does appear that AVE is
converging to a local minimum of E^N. As in example S2.2 we
suspect that the IDN problem for AVE suffers a lack of identifiability.
The IDN problem for SPLINE seems to be much better behaved.

In order to further investigate identifiability for problems
with unknown initial data we essentially repeated this example
with identical dynamics, changing only the initial data to

$$\eta = 1 , \quad \varphi(s) = 1 + s , \quad -1 \leq s < 0 .$$

Using the same start-ups as above we found that SPLINE converged
for all N values, whereas AVE never did. Results are summarized
in Table S2.4.2.

AVE			SPLINE		
N	$\hat{a}_1^N$	$\|z*(0)-\hat{z}^N(0)\|$	N	$\hat{a}_1^N$	$\|z*(0)-\hat{z}^N(0)\|$
2	-4.4103	2.08	2	-4.4382	.1595
4	-4.9924	4.53	4	-3.9381	.0867
8	-4.2651	41.76	8	-4.0031	.0287
16	did not converge		16	-4.0031	.0201
32	did not converge		32	-4.0001	.0386

TABLE S2.4.1

AVE			SPLINE		
N	$\hat{a}_1^N$	$\|z*(0)-\hat{z}^N(0)\|$	N	$\bar{a}_1^N$	$\|z*(0)-\hat{z}^N(0)\|$
2			2	-4.5201	.0563
4	did not converge		4	-4.0975	.0318
8			8	-4.0282	.0123
16			16	-4.0123	.0193
32			32	-4.0122	.0936

TABLE S2.4.2

(linear initial data)

N = 4   ESTIMATE OF INITIAL DATA

$\dot{x}(t) = .05x(t) + a_1 x(t-1) + u_{.1}(t)$

$x(0) = \eta$, $x_0(s) = \varphi(S)$

$\hat{\varphi}_A^4$

ϕ_S^4

φ^*

$\hat{\varphi}_S^4$ $\hat{\eta}_S^4$

φ^* $\hat{\eta}_A^4$

TIME
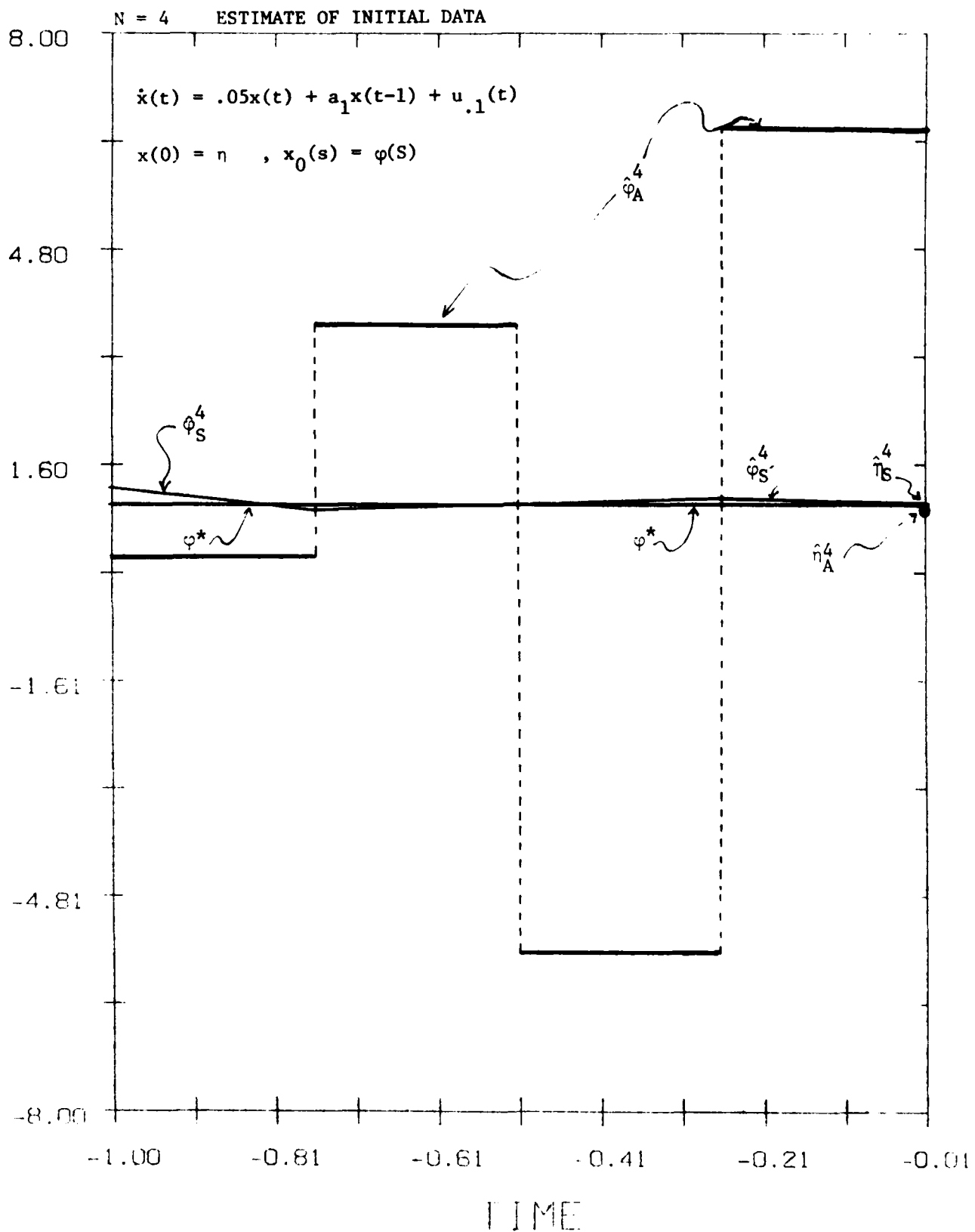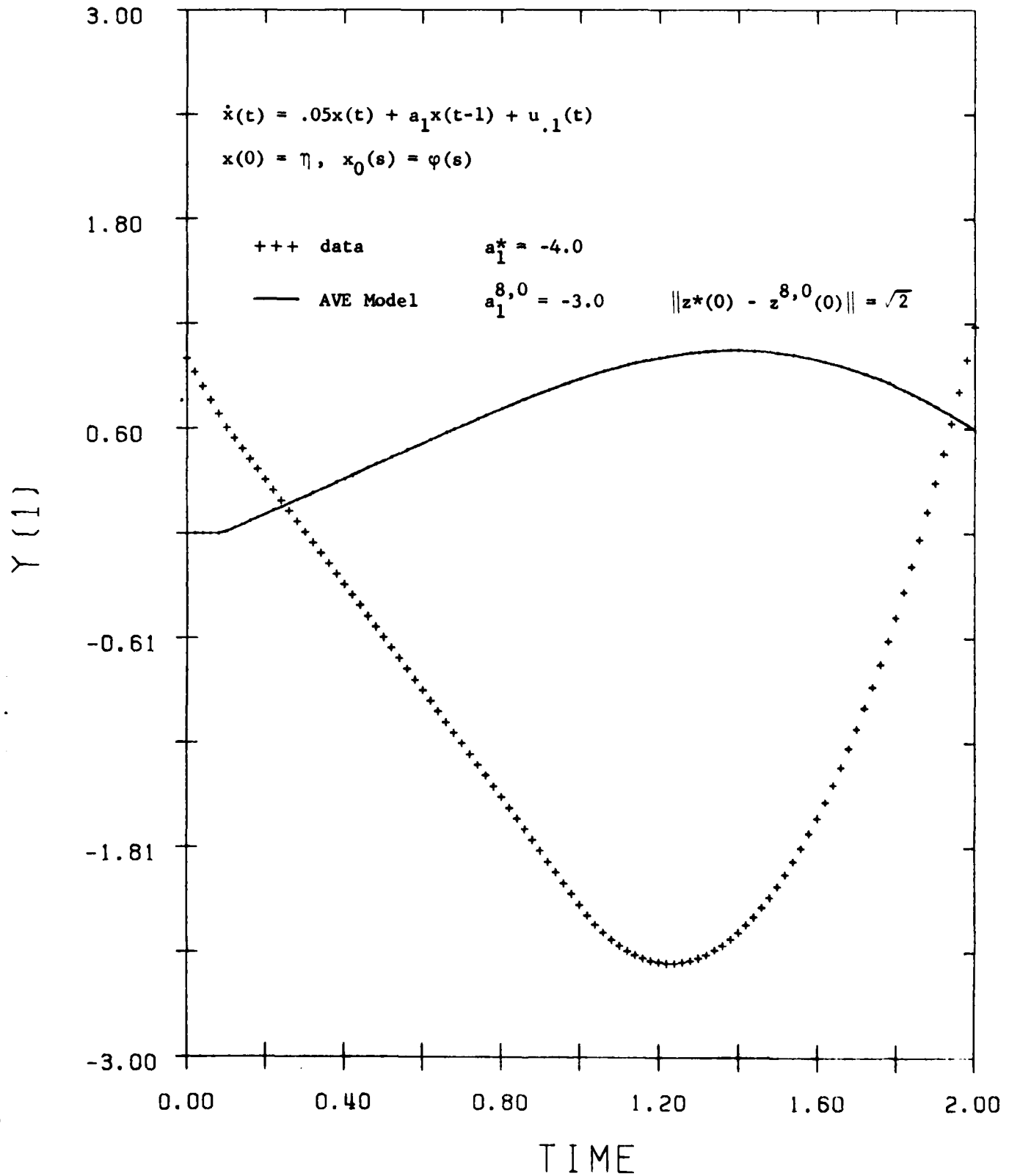
FIGURE S2.4.1
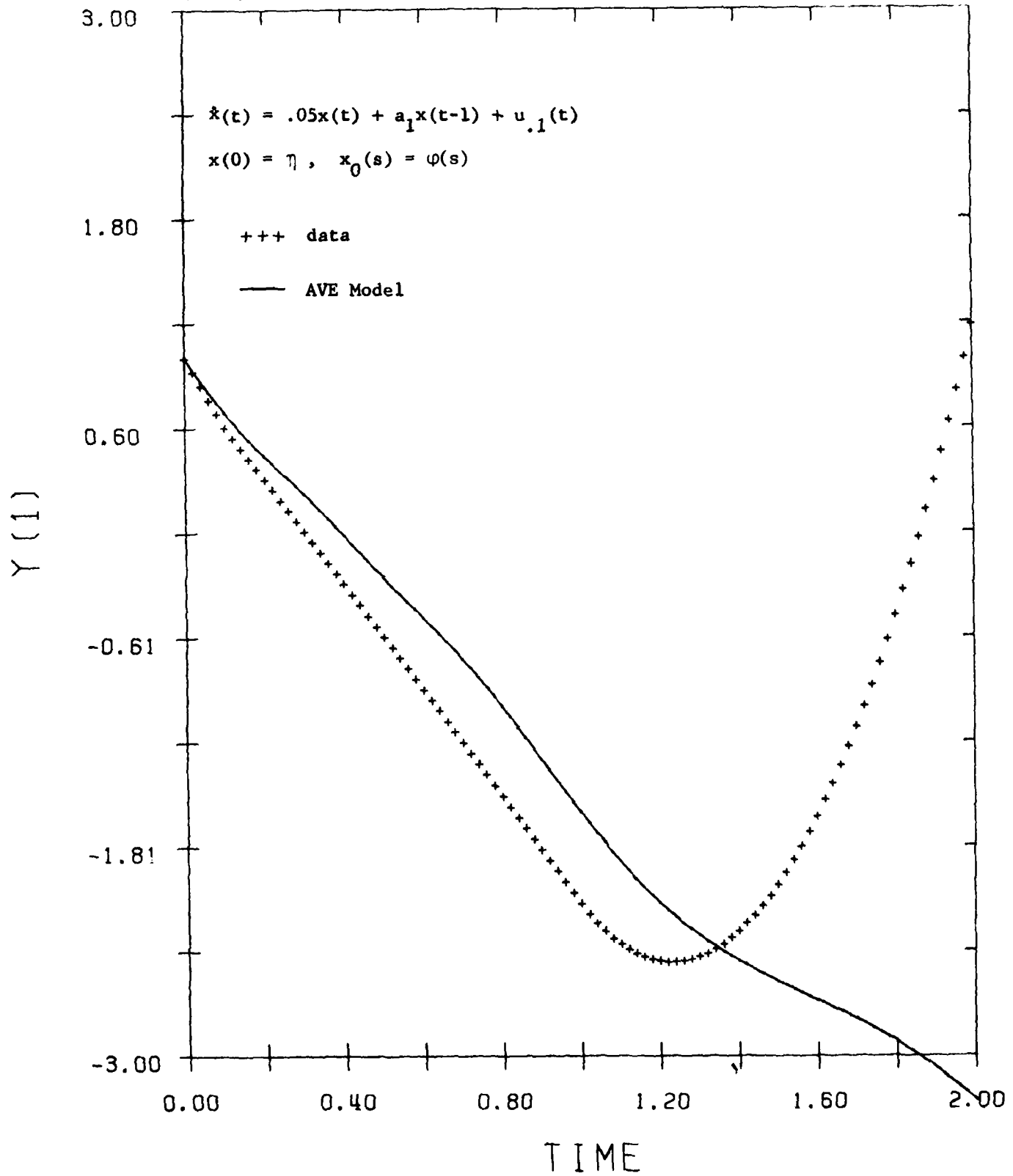
FIGURE S2.4.2

FIGURE S2.4.3

S2.4N8AV   ITR= 9

$$\dot{x}(t) = .05x(t) + a_1 x(t-1) + u_{.1}(t)$$

$$x(0) = \eta \ , \quad x_0(s) = \varphi(s)$$

+++ data $a_1^* = -4.00$

——— AVE Model $a_1^{8,9} = -4.2651$ $\|z^*(0) - z^{8,9}(0)\| = 41.76$
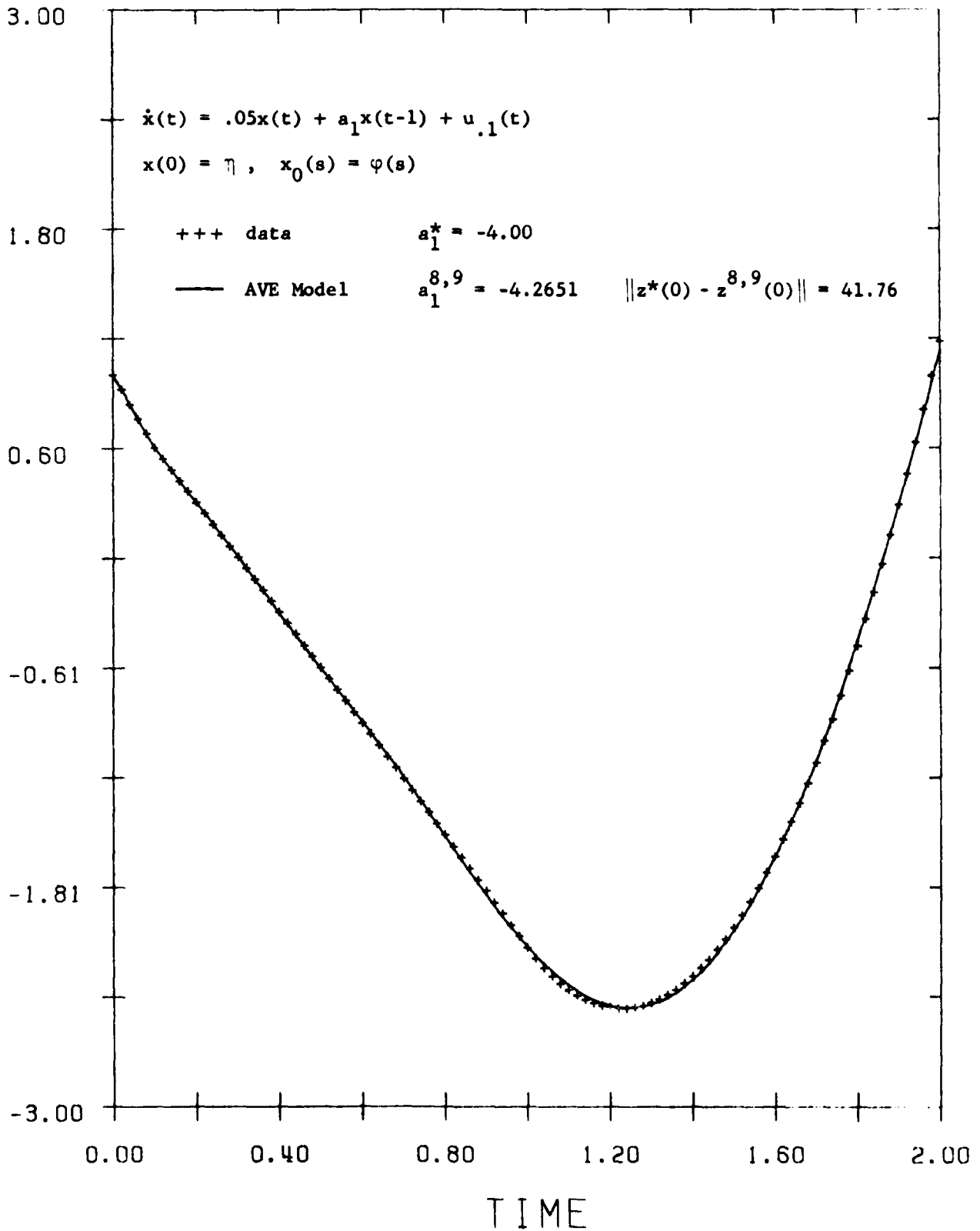
FIGURE S2.4.4

S2.4N8SP   ITR= 0

$$\dot{x}(t) = .05x(t) + a_1 x(t-1) + u_{.1}(t)$$

$$x(0) = \eta, \quad x_0(s) = \varphi(s)$$

+ + + data $a_1^* = -4.0$

——— SPLINE Model $a_1^{8,0} = -3.0$ $\|z^*(0) - z^{8,0}(0)\| = \sqrt{2}$

Y(1)

TIME
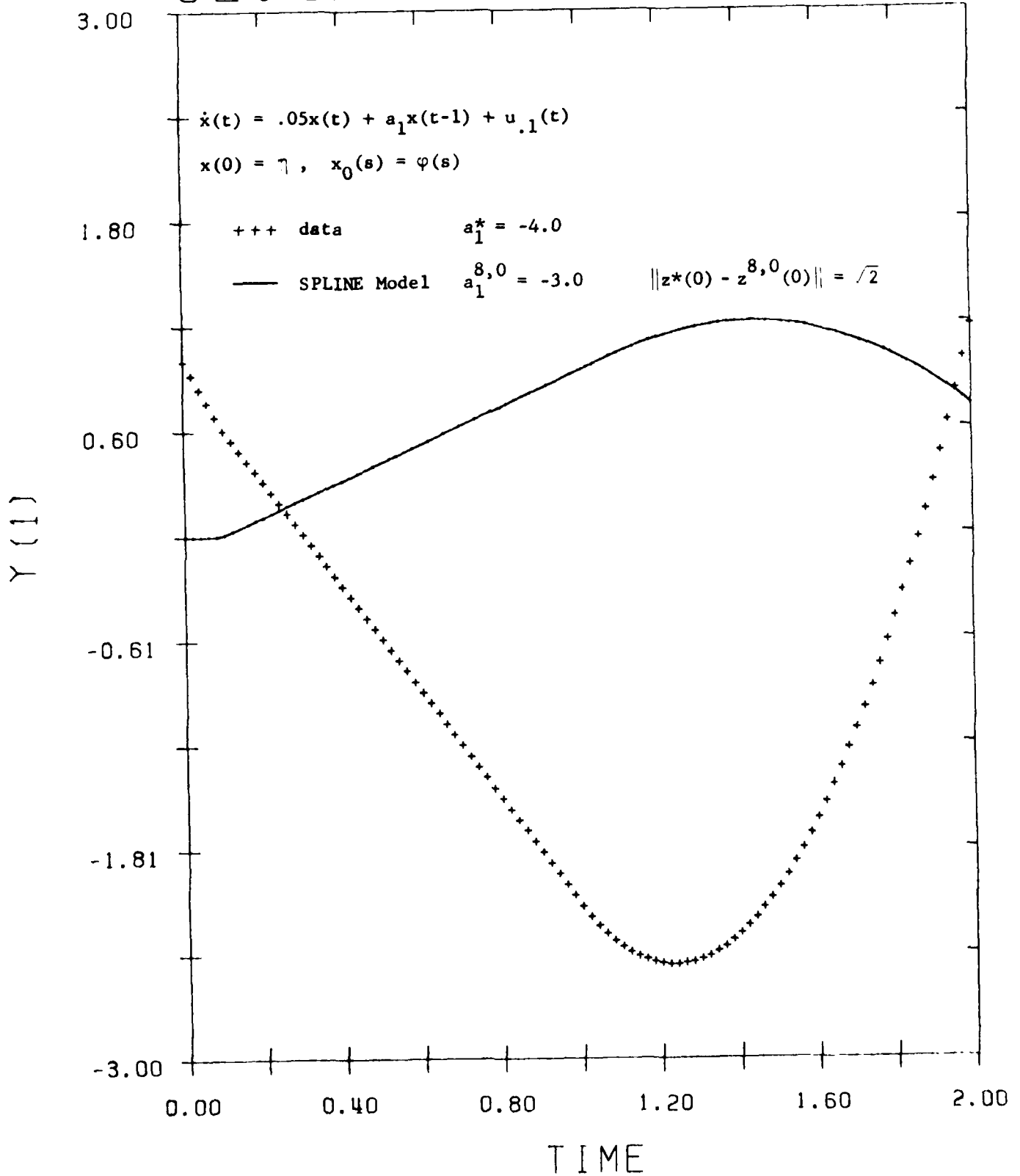
FIGURE S2.4.5

S2.4N8SP   ITR= 4

$\dot{x}(t) = .05x(t) + a_1 x(t-1) + u_{.1}(t)$

$x(0) = \eta$, $x_0(s) = \varphi(s)$

+ + +   data

———   SPLINE Model

3.00

1.80

0.60

-0.61

-1.81

-3.00

Y(1)

0.00   0.40   0.80   1.20   1.60   2.00

TIME
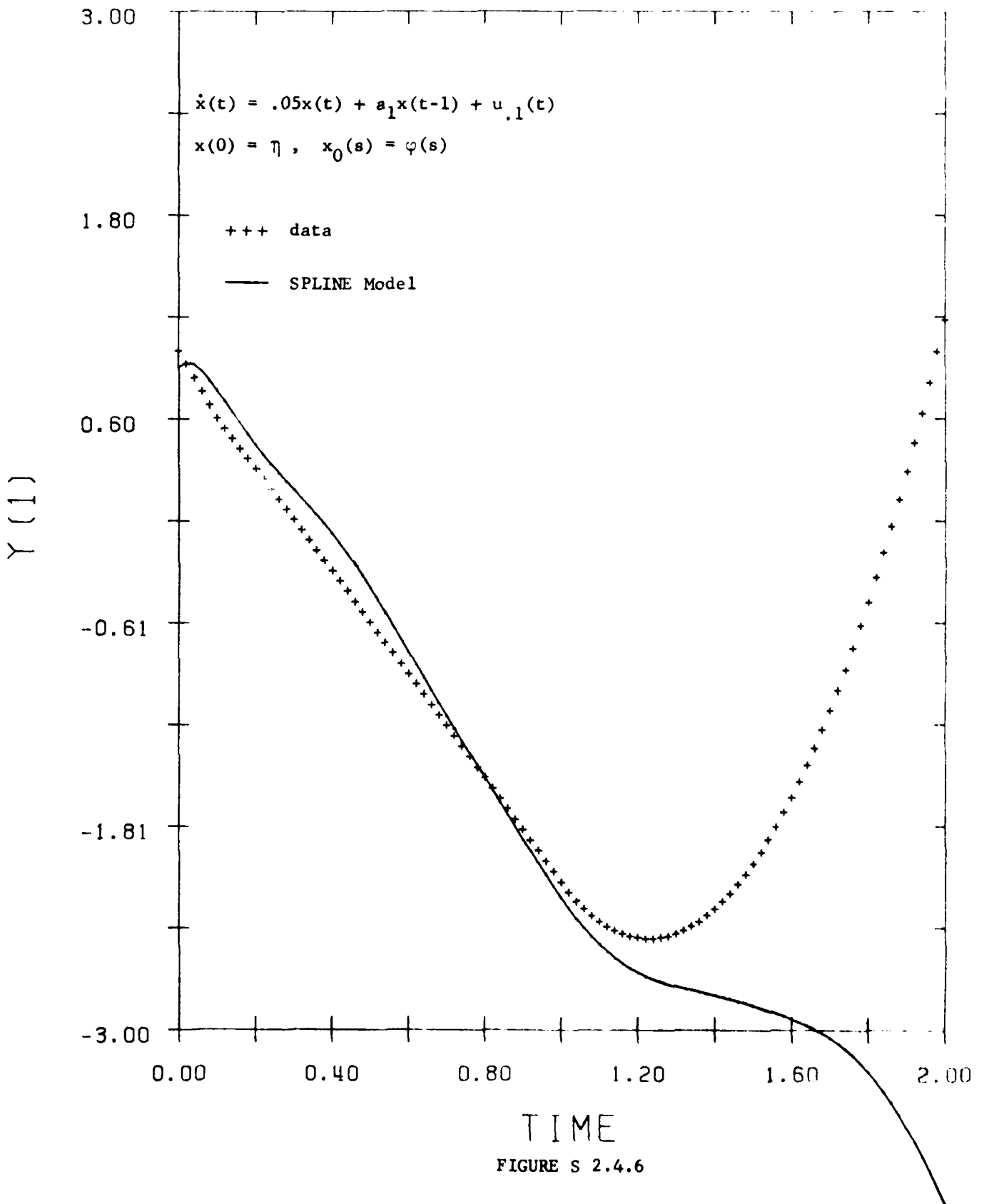
FIGURE S 2.4.6

S2.4N8SP   ITR= 9

$\dot{x}(t) = .05x(t) + a_1 x(t-1) + u_{.1}(t)$

$x(0) = \eta$, $x_0(s) = \varphi(s)$

+ + + data $a_1^* = -4.0$

—— SPLINE Model $a_1^{8,9} = -4.0031$ $\|z^*(0) - z^{8,9}(0)\| = .0287$
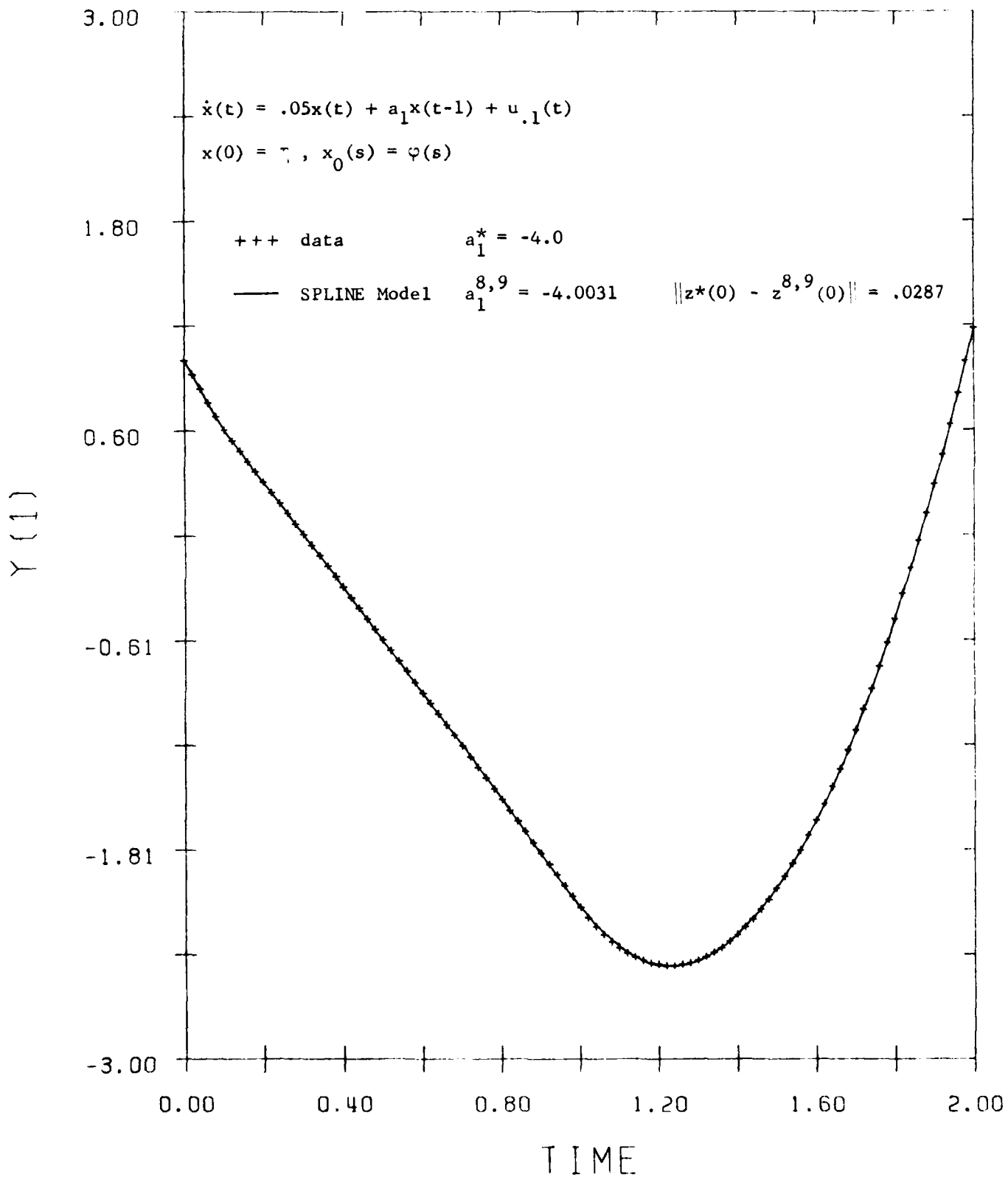
FIGURE S2.4.7

ID MODEL 01

This model describes a (reasonably realistic) mechanical oscillator with retarded restoring and retarded damping forces. The system is governed by the second order equation

$$\ddot{x}(t) + 36x(t) + 2.5\,\dot{x}(t-1) + 9x(t-1) = u_{.1}(t)$$

with initial data

$$x_0(s) \equiv 1 \ , \quad \dot{x}_0(s) \equiv 0 \ , \ -1 \le s \le 0 \ ,$$

and scalar output (which represents position)

$$y(t) = x(t) \ .$$

This second order equation is equivalent to the two dimensional system

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -36 & 0 \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -9 & -2.5 \end{bmatrix}\begin{bmatrix} x_2(t-1) \\ x_2(t-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_{.1}(t) \ ,$$

with initial condition

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_0 (s) \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix} \ , \ -1 \le s \le 0 \ ,$$

and output

$$y(t) = [1 \; 0] \begin{bmatrix} x_1(t) \\ \\ x_2(t) \end{bmatrix}.$$

This system was integrated forward (using the method of steps) to

obtain the analytic solution on $[0,2]$. Again, data was generated

at 101 equally spaced points by evaluating the true solution. This

data was used in the following examples; 01.1 - 01.2.

EXAMPLE 01.1

This experiment is devoted to the estimation of the three coefficients in the model. Therefore, we assume our model is described by the second order equation

$$\ddot{x}(t) + \omega^2 x(t) + a_0 \dot{x}(t-1) + a_1 x(t-1) = u_{.1}(t)$$

with initial data

$$x_0(s) \equiv 1 \quad , \quad \dot{x}_0(s) \equiv 0 \quad , \quad -1 \le s \le 0 \quad ,$$

and output

$$y(t) = x(t) \quad .$$

The problem is to estimate ω, a_0 and a_1.

Since the basic system is two dimensional for each N the approximating systems for AVE and SPLINE is of dimension $2 \cdot (N + 1)$. In order to keep the program size reasonably small (our objective is to test the algorithms and not to develop computer codes) we solved the approximating identification problems for N = 2, 4, 8 and 16. This allowed us to use the same code for scalar and two dimensional systems without increasing the "size" of the code; therefore keeping the computing cost minimal.

The start-up values for $\omega^* = 6$, $a_0^* = 2.5$ and $a_1^* = 9$ were

$$\omega^{N,0} = 5.0 \quad , \quad a_0^{N,0} = 1.0 \quad , \quad a_1^{N,0} = 5.0 \quad .$$

Tables 01.1.1 and 01.1.2 show the parameter estimates for AVE and SPLINE, respectively. Observe that the N = 16 estimates produced by the AVE procedure are such that the "relative ℓ_1 error" ($\frac{|e_N|}{|\gamma^*|}$) is approximately 20%. On the other hand the N = 16 SPLINE estimate has relative ℓ_1 error of less than 1%.

Figures 01.1.1 - 10.1.4 compare the data fits for N = 2. The start-ups (ITR = 0) are shown as well as the converged fits (ITR = 10 for AVE and ITR = 14 for SPLINE). Figures 01.1.5 - 01.1.6 show the converged data fits for N = 16. Note that in this case both AVE and SPLINE converged after 4 iterations of the MLE algorithm. However, the SPLINE procedure provided a near perfect data fit.

This example is typical of most of the vector systems that were studied. Generally speaking, the SPLINE algorithm produced better parameter estimates and data fits.

AVE			
N	$\hat{\omega}^N$	$\hat{a}_0^N$	$\hat{a}_1^N$	$|e_N|$
2	6.3864	-12.8383	4.2478	20.4769
4	5.7480	- 5.4170	7.3614	9.8076
8	5.6564	- 1.8301	9.7648	5.4385
16	5.7873	3.6873	6.6713	3.7287
$\gamma^* =$	6.0000	2.5000	9.0000	

TABLE 01.1.1

SPLINE				
N	$\hat{\omega}^N$	$\hat{a}_0^N$	$\hat{a}_1^N$	$|e_N|$
2	6.1102	-5.7950	10.3718	9.777
4	6.4861	5.6291	13.2680	7.8832
8	6.0432	2.8791	9.2921	.7144
16	6.0079	2.5761	9.0591	.1431
$\gamma^* =$	6.0000	2.5000	9.0000	

TABLE 01.1.2

01.1N2AV   ITR= 0

$$\ddot{x}(t) + \omega^2 x(t) + a_0\dot{x}(t-1) + a_1 x(t-1) = u_{.1}(t)$$

+ + +   data $\omega^* = 6.0$ $a_0^* = 2.5$ $a_1^* = 9.0$

———   AVE Model $\omega^{2,0} = 5.0$ $a_0^{2,0} = 1.0$ $a_1^{2,0} = 5.0$

TIME

FIGURE 01.1.1

FIGURE 01.1.2

Ø1.1N2SP   ITR= 0

$$\ddot{x}(t) + \omega^2 x(t) + a_0\dot{x}(t-1) + a_1 x(t-1) = u_{.1}(t)$$

+++ data $\omega^* = 6.0$ $a_0^* = 2.5$ $a_1^* = 9.0$

—— SPLINE Model $\omega^{2,0} = 5.0$ $a_0^{2,0} = 1.0$ $a_1^{2,0} = 5.0$

Y(1)
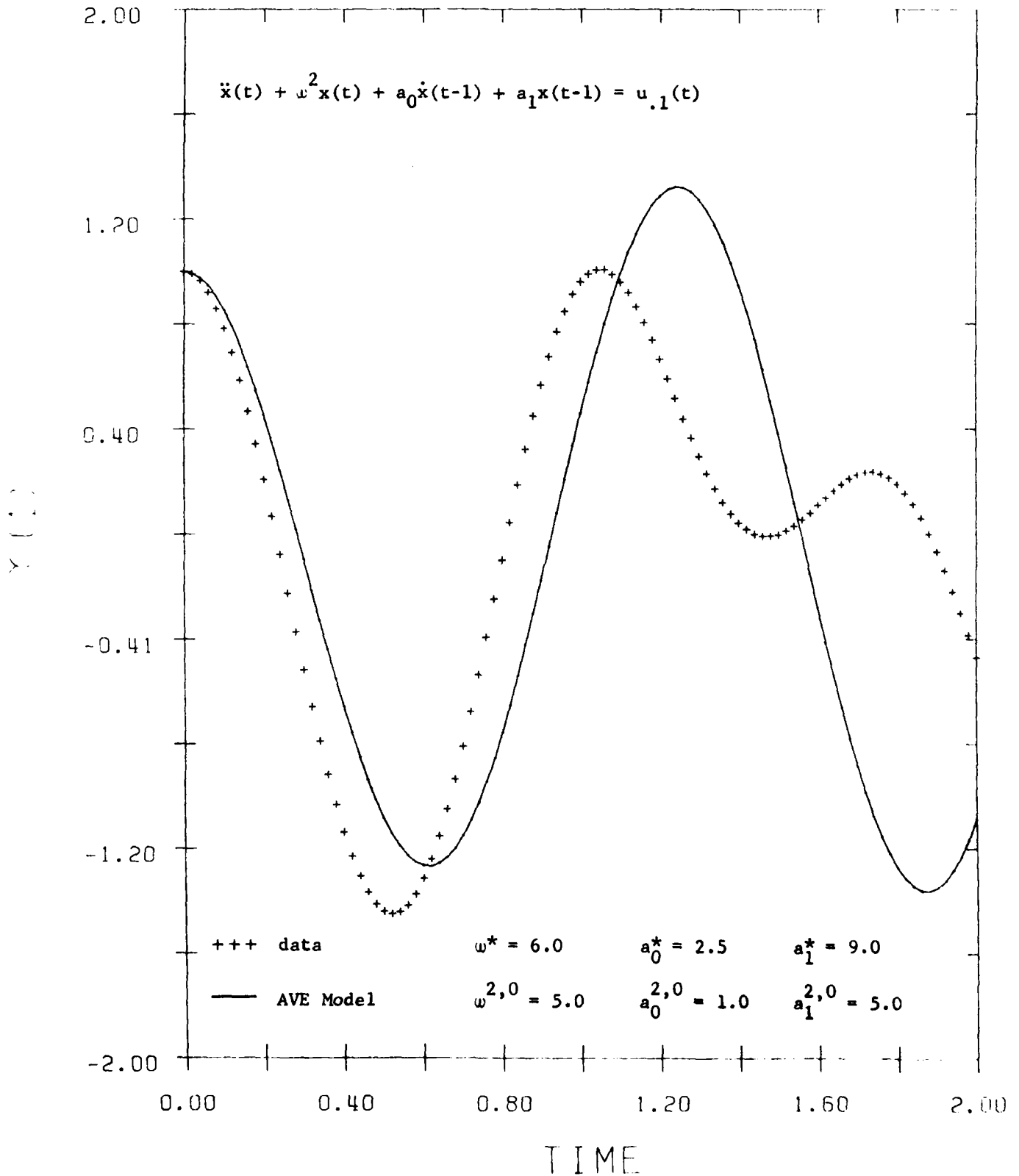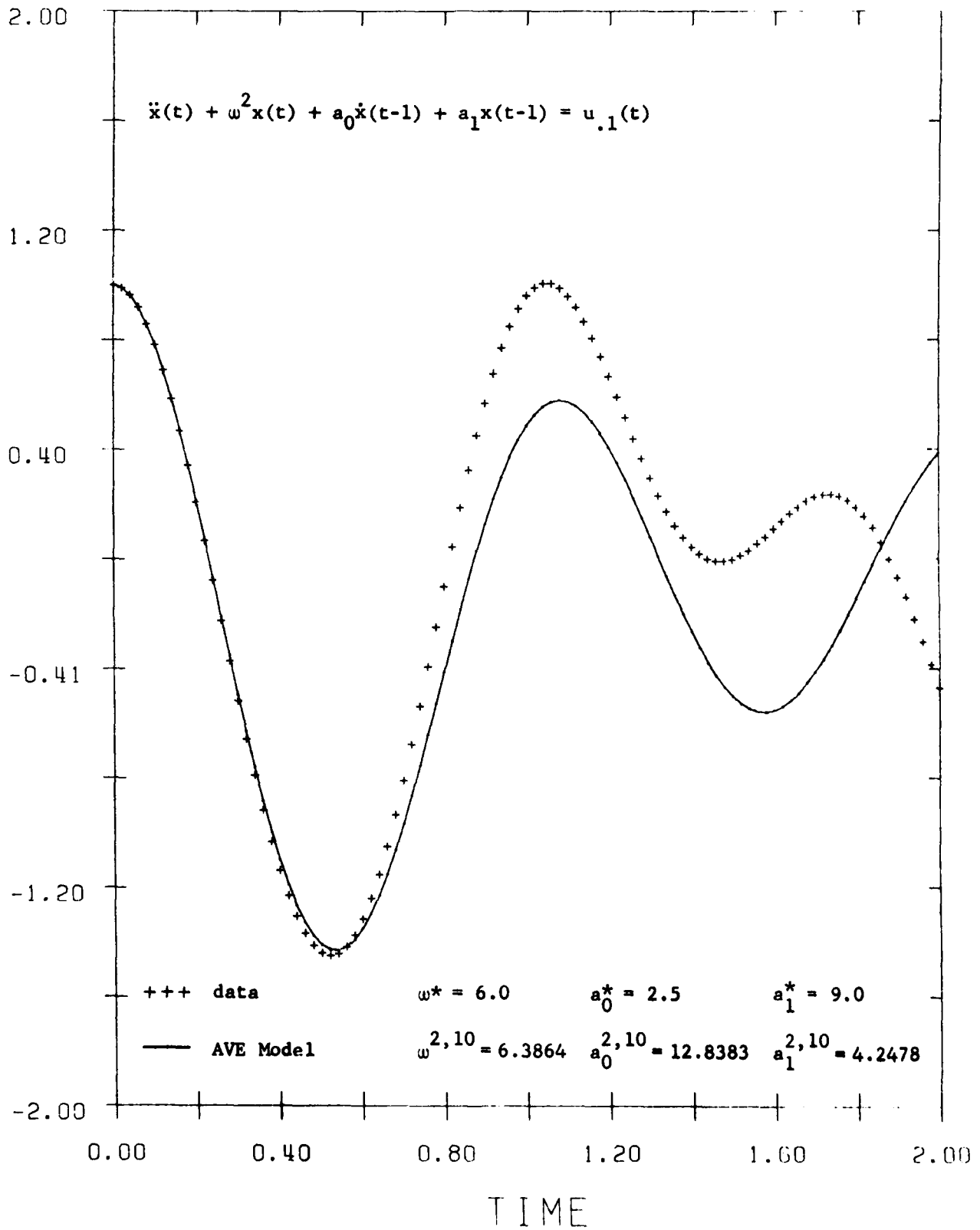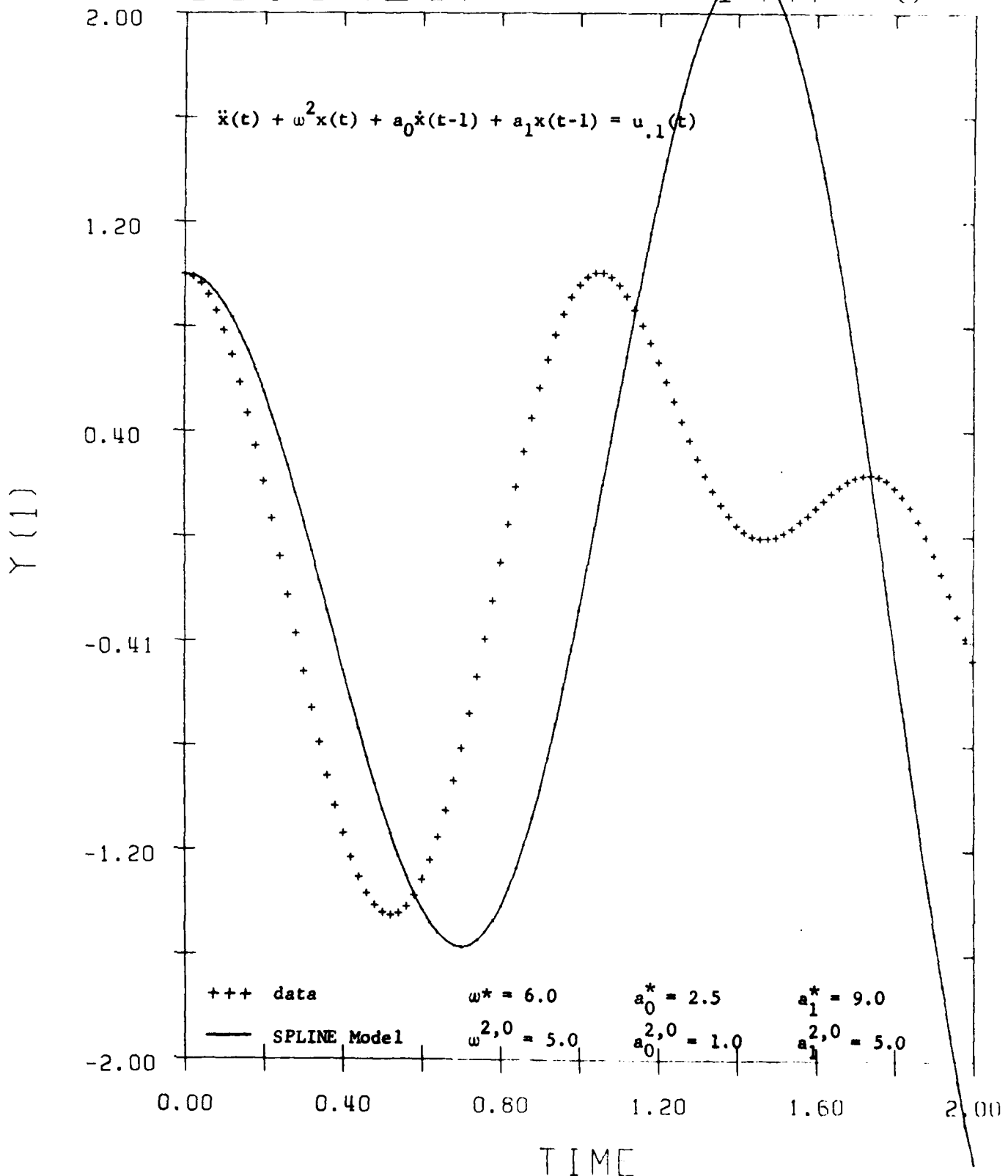
TIME

FIGURE 01.1.3

01.1N2SP   ITR= 14

$$\ddot{x} + \omega^2 x(t) + a_0 \dot{x}(t-1) + a_1 x(t-1) = u_{.1}(t)$$

+++ data $\omega^* = 6.0$ $a_0^* = 2.5$ $a_1^* = 9.0$

——— SPLINE Model $\omega^{2,14} = 6.1102$ $a_0^{2,14} = -5.7950$ $a_1^{2,14} = 10.3718$
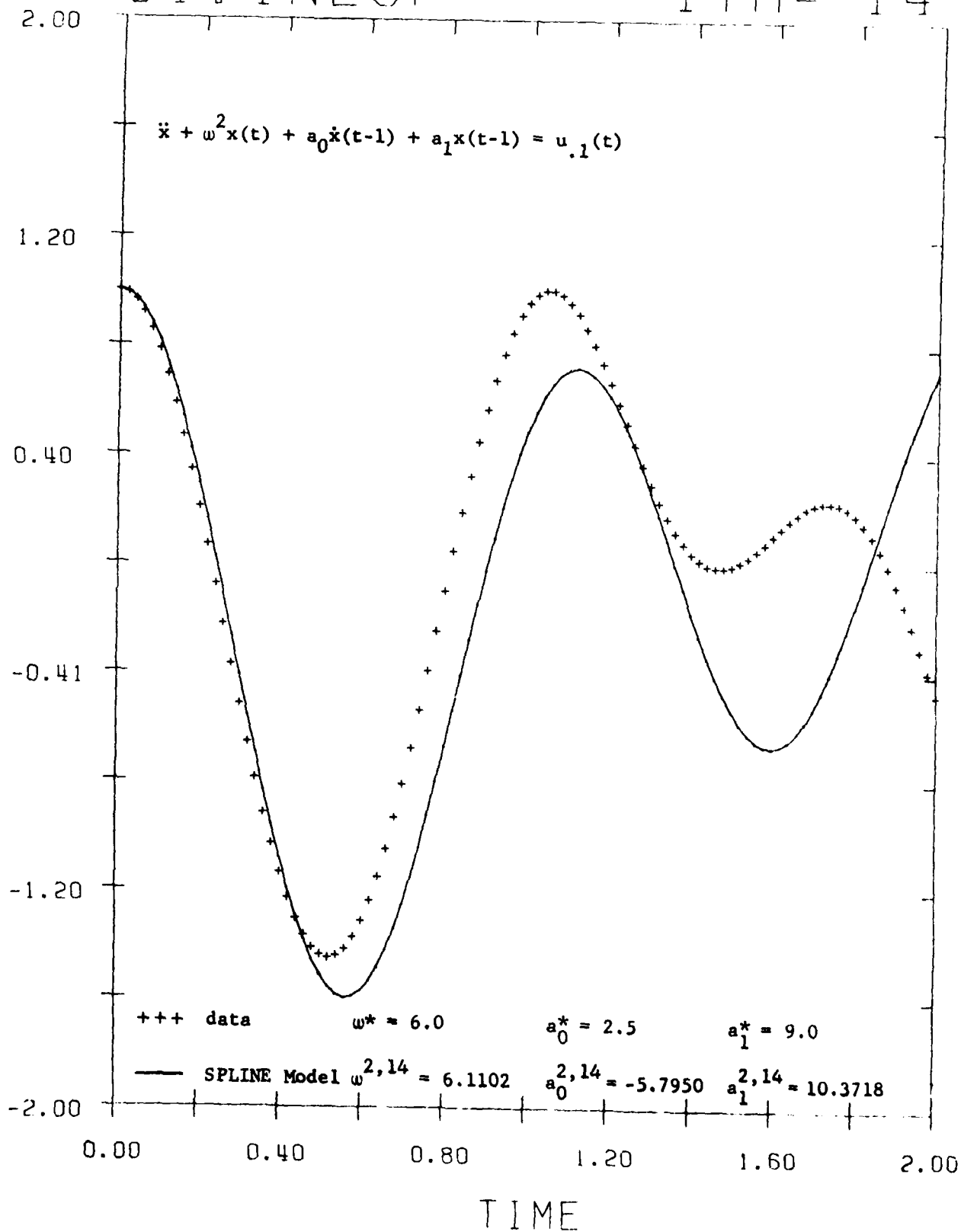
TIME

FIGURE 01.1.4

01.1N16A   ITR- 4

$$\ddot{x}(t) + \omega^2 x(t) + a_0 \dot{x}(t-1) + a_1 x(t-1) = u_{.1}(t)$$

+++ data $\omega* = 6.0$ $a_0^* = 2.5$ $a_1^* = 9.0$

——— AVE Model $\omega^{16,4} = 5.7873$ $a_0^{16,4} = 3.6873$ $a_1^{16,4} = 6.6713$

TIME

FIGURE 01.1.5

O1.1N16S   ITR= 4

$$\ddot{x}(t) + \omega^2 x(t) + a_0\dot{x}(t-1) + a_1 x(t-1) = u_{.1}(t)$$

+++ data $\omega^* = 6.0$ $a_0^* = 2.5$ $a_1^* = 9.0$

——— SPLINE Model $\omega^{16,4} = 6.0079$ $a_0^{16,4} = 2.5761$ $a_1^{16,4} = 9.0591$
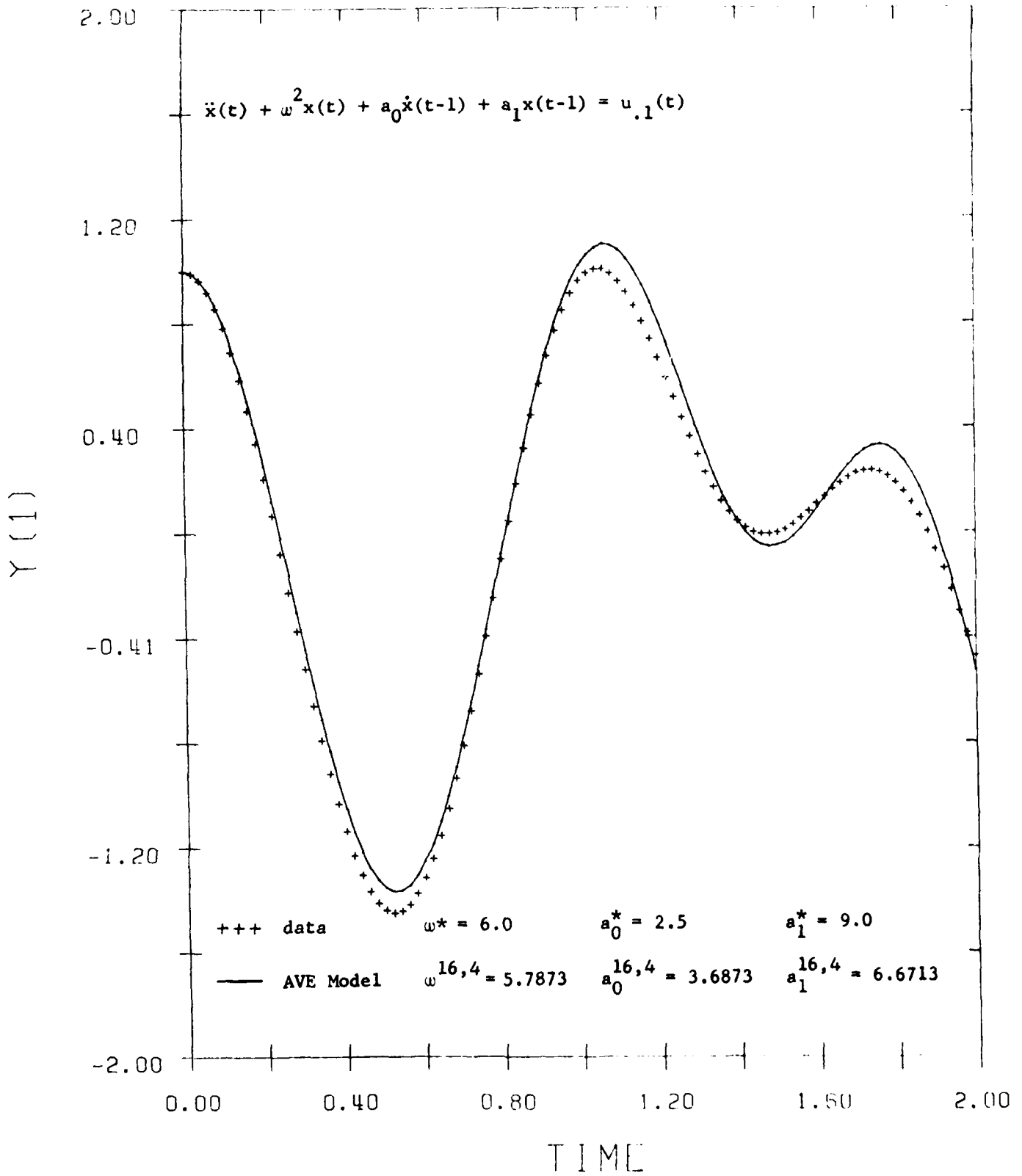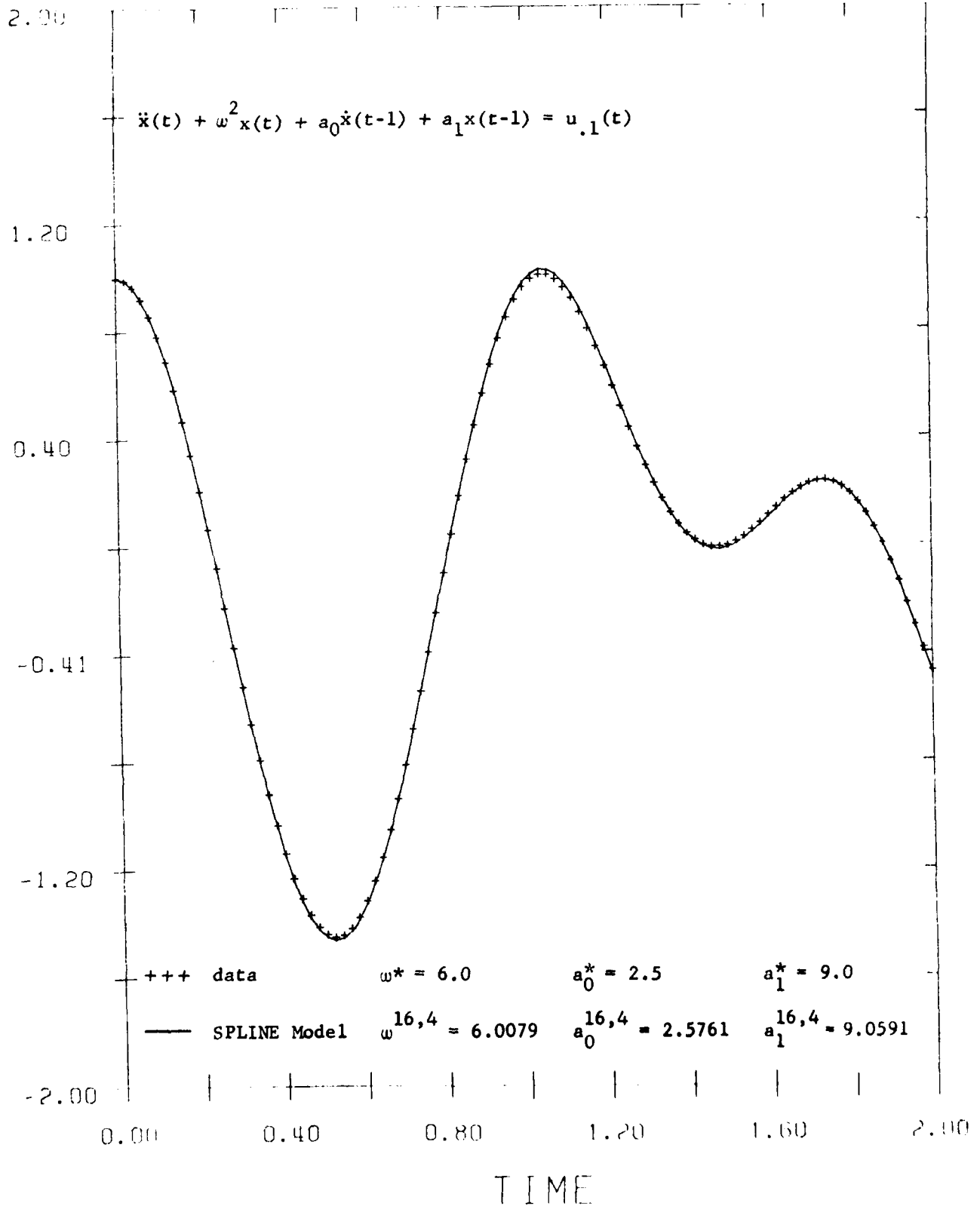
TIME

FIGURE 01.1.6

EXAMPLE 01.2

In this example we identify only the time delay r. The system is assumed to be governed by the model

$$\ddot{x}(t) + 36x(t) + 2.5 \dot{x}(t-r) + 9x(t-r) = u_{.1}(t) \ ,$$

with initial data

$$x_0(s) \equiv 1 \ , \ \dot{x}_0(s) \equiv 0 \ , \ -r \le s \le 0 \ ,$$

and scalar output

$$y(t) = x(t) \ ,$$

where r is the unknown delay.

This example proved to be very interesting. The start-up $r* = 1.0$ was taken to be

$$r^{N,0} = 1.2 \ .$$

At $N = 2$, the AVE procedure did not converge (in fact estimates for $\hat{r}^2$ were growing without bound), while the SPLINE algorithm converged to the estimate $\hat{r}^2 = 2.3476$. At $N = 4$, SPLINE produced the estimate $\hat{r}^4 = .9830$. However, for $N = 4$, AVE converged to the estimate of $\hat{r}^4 = 4.8694$. For higher N, the SPLINE procedure produced better and better estimates. At $N = 8$ the AVE scheme produced a sequence of MLE estimates that oscillated between the values .7000 and 1.3000. More precisely, the MLE iterations continued to

produce sequences similiar to

$$\cdots .7113 , 1.0477 , 1.2755 , \cdots , .7082 , 1.0396 , 1.2811 \cdots .$$

Consequently, the N = 8 AVE scheme never "converged"! At N = 16, both AVE and SPLINE converged to reasonable estimates of the parameter r.

This example was repeated using a start-up value of $r^{N,0} = .8$ and the results were exactly the same. Table 01.2.1 contains a summary of the convergence for this example. Figures 01.2.1 - 01.2.4 illustrate the start-ups and converged data fits at N = 16 for AVE and SPLINE.

AVE			SPLINE		
N	$\hat{r}^N$	$|e_N|$	N	$\hat{r}^N$	$|e_N|$
2	did not converge		2	2.3476	3.3476
4	-4.8694	5.8694	4	.9830	.0170
8	did not converge		8	.9939	.0061
16	.9274	.0726	16	.9987	.0013
r* =	1.0000		r* =	1.0000	

TABLE 01.2.1
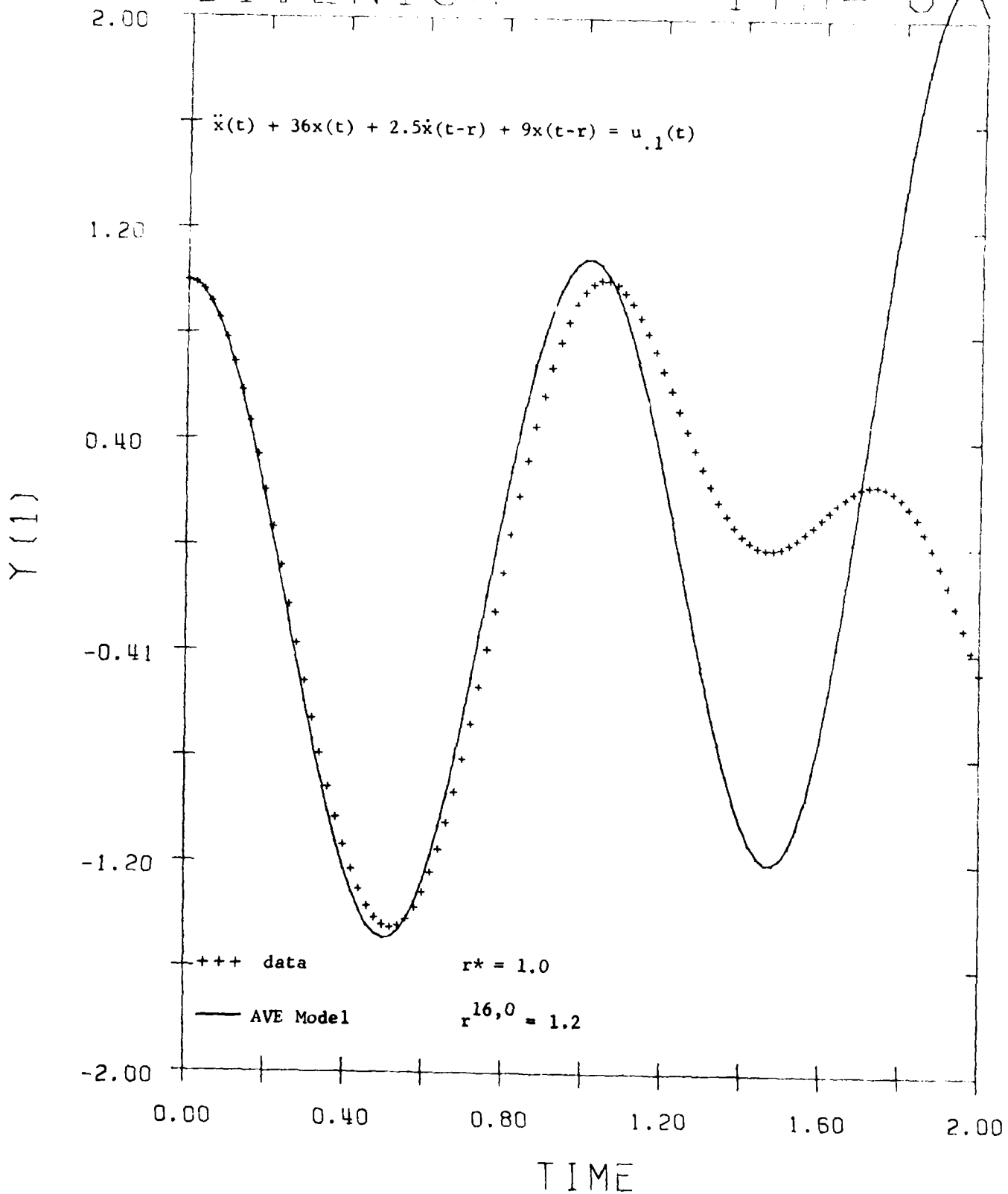
FIGURE 01.2.1

01.2N16A   ITR= 4

$$\ddot{x}(t) + 36x(t) + 2.5\dot{x}(t-r) + 9x(t-r) = u_{.1}(t)$$

+++ data   r* = 1.0

—— AVE Model $r^{16,4}$ = .9274

Y(1)

TIME
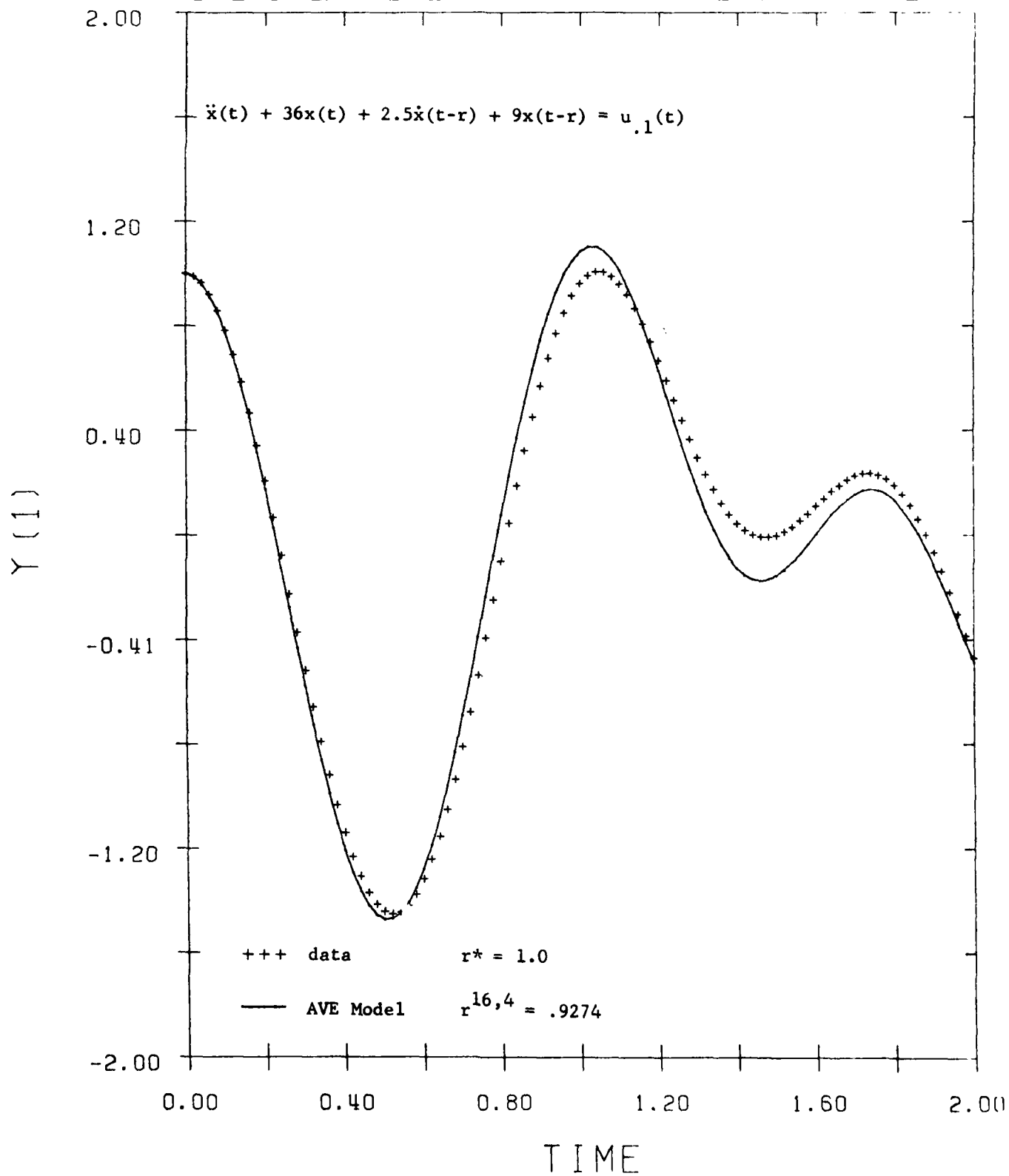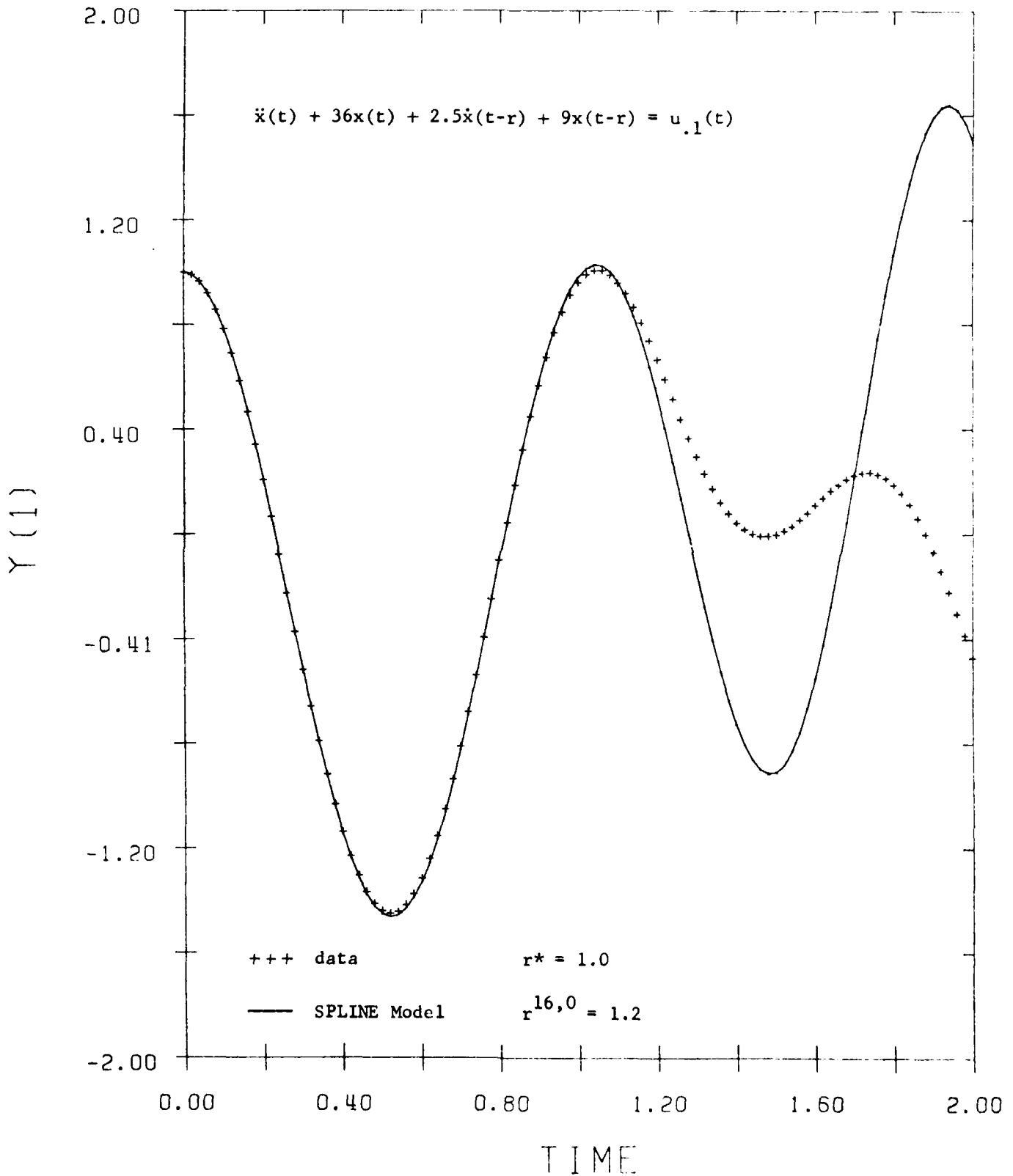
FIGURE 01.2.2
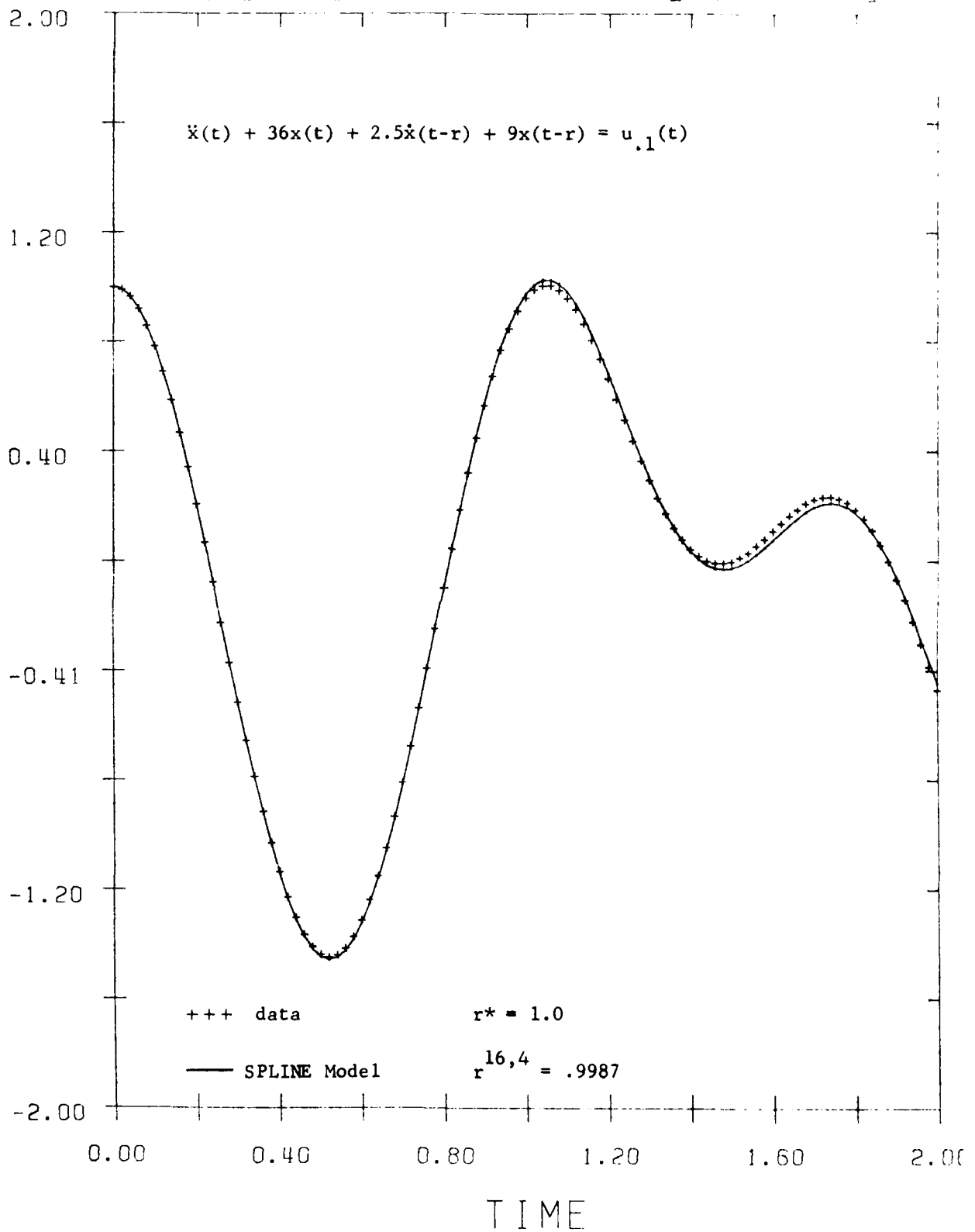
FIGURE 01.2.3

FIGURE 01.2.4

ID MODEL 02

This model is the oscillator governed by the equation

$$\ddot{x}(t) + 16x(t) + 10\dot{x}(t-1) - 10x(t-1) = u_{.1}(t) \ ,$$

with initial data

$$x_0(s) \equiv 1 \ , \quad \dot{x}_0(s) \equiv 0 \ , \quad -1 \le s \le 0 \ ,$$

and scalar output

$$y(t) = x(t) \ .$$

As before, data was generated at 101 equally spaced points by solving the system analytically and evaluating the solution. This data was used in the following examples; 02.1 - 02.2.

EXAMPLE 02.1

Here we identify all three of the equation coefficients. The model is described by

$$\ddot{x}(t) + \omega^2 x(t) + a_0 \dot{x}(t-1) + a_1 x(t-1) = u_{.1}(t)$$

with initial data

$$x_0(s) \equiv 1 \ , \quad \dot{x}_0(s) \equiv 0 \ , \quad -1 \le s \le 0 \ ,$$

and output

$$y(t) = x(t) \ .$$

We seek to estimate the true parameters $\omega^* = 4.0$, $a_0^* = 10.0$ and $a_1^* = -10.0$. Start-up values for these parameters were selected to be

$$\omega^{N,0} = \sqrt{20} \ , \quad a_0^{N,0} = 0.0 \ , \quad a_1^{N,0} = -9.0 \ .$$

Runs were made for $N = 2, 4, 8$ and 16. The results for AVE and SPLINE are summarized in Tables 02.1.1 and 02.1.2, respectively. Again, the error $|e_N|$ is taken to be in the ℓ_1 norm, and the relative ℓ_1 error at $N = 16$ is about 3% for AVE and less than 1% for SPLINE.

Figures 02.1.1 - 02.1.2 show the $N = 16$ AVE data fits for the start-ups and converged values of the parameters. Figures 02.1.3 - 02.1.4 show the $N = 16$ SPLINE data fits for the start-ups and

converged values of the parameters. The SPLINE procedure clearly
does better.

In this example (as well as others) we checked the CPU times
required for each run. The two schemes AVE and SPLINE basically
require the same amount of computer time for each iteration of the
MLE algorithm. For example, at N = 16, the AVE scheme used approx-
imately 17.75 sec/ITR while the SPLINE scheme used approximately
18.40 sec/ITR. Such figures are typical of all the runs.

AVE			
N	$\hat{\omega}^N$	$\hat{a}_0^N$	$\hat{a}_1^N$	$|e_N|$
2	did	not	converge	___
4	did	not	converge	___
8	3.4386	12.3634	-6.6389	.7128
16	3.9826	10.4641	-9.7997	.6818
$\gamma^* =$	4.0000	10.0000	-10.0000	

TABLE 02.1.1

	SPLINE			
N	$\hat{\omega}^N$	$\hat{a}_0^N$	$\hat{a}_1^N$	$|e_N|$
2	3.8092	9.0371	-9.3642	1.7895
4	3.9751	9.9323	-9.9241	.1685
8	3.9963	9.9511	-9.9978	.0548
16	3.9943	9.9920	-9.9812	.0325
$\gamma^* =$	4.0000	10.0000	-10.0000	

TABLE 02.1.2

02.1N16A   ITR= 0

$$\ddot{x}(t) + \omega^2 x(t) + a_0 \dot{x}(t-1) + a_1 x(t-1) = u_{.1}(t)$$

+ + + data $\omega^* = 4.0$ $a_0^* = 10.0$ $a_1^* = -10.0$

—— AVE Model $\omega^{16,0} = \sqrt{20}$ $a_0^{16,0} = 0.0$ $a_0^{16,0} = -9.0$

FIGURE 02.1.1

02.1N16A   ITR= 3

$\ddot{x}(t) + \omega^2 x(t) + a_0\dot{x}(t-1) + a_1 x(t-1) = u_{.1}(t)$

+++ data $\omega^* = 4.0$ $a_0^* = 10.0$ $a_0^* = -10.0$

——— AVE Model $\omega^{16,3} = 3.9826$ $a_0^{16,3} = 10.4641$ $a_1^{16,3} = -9.7997$
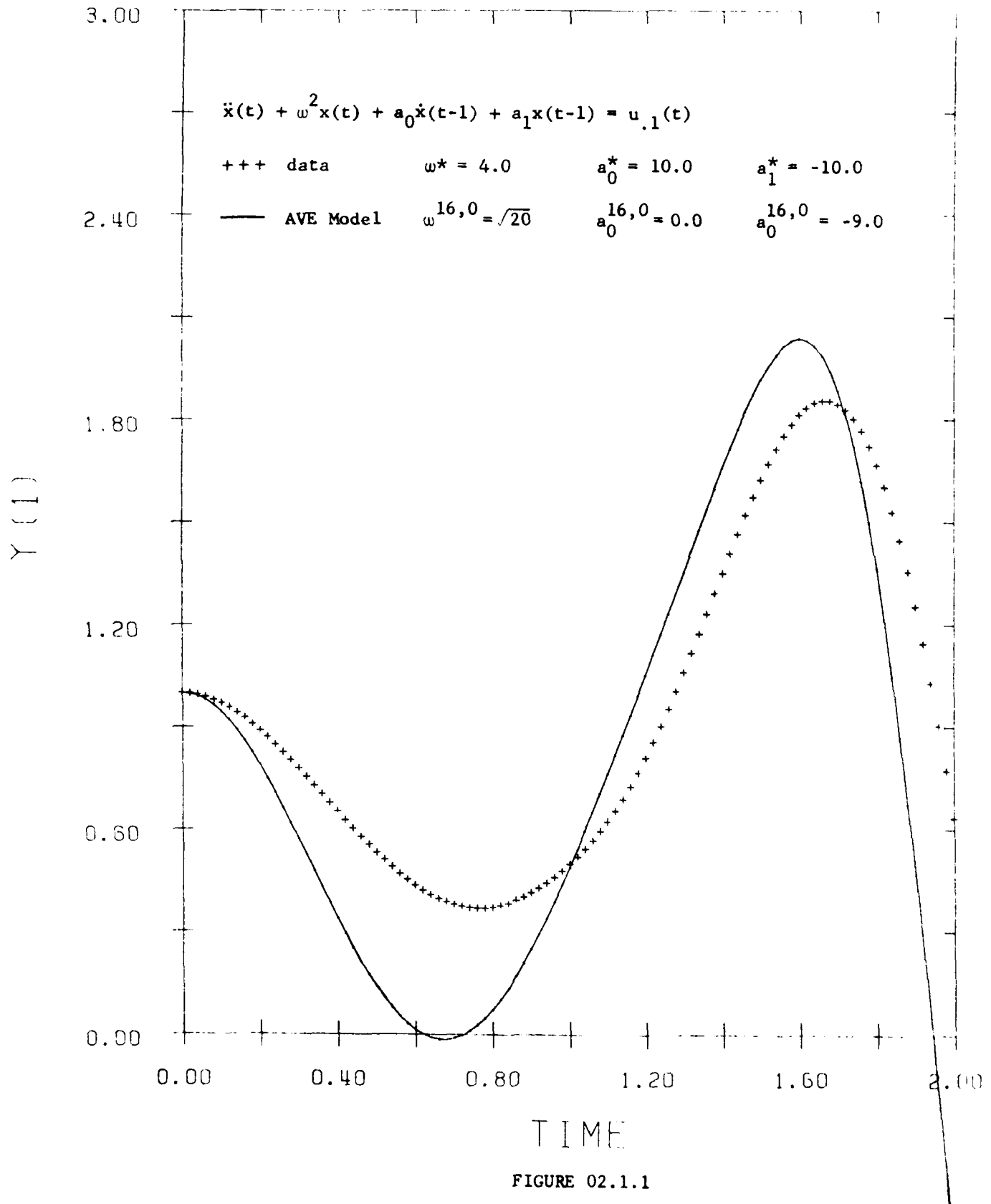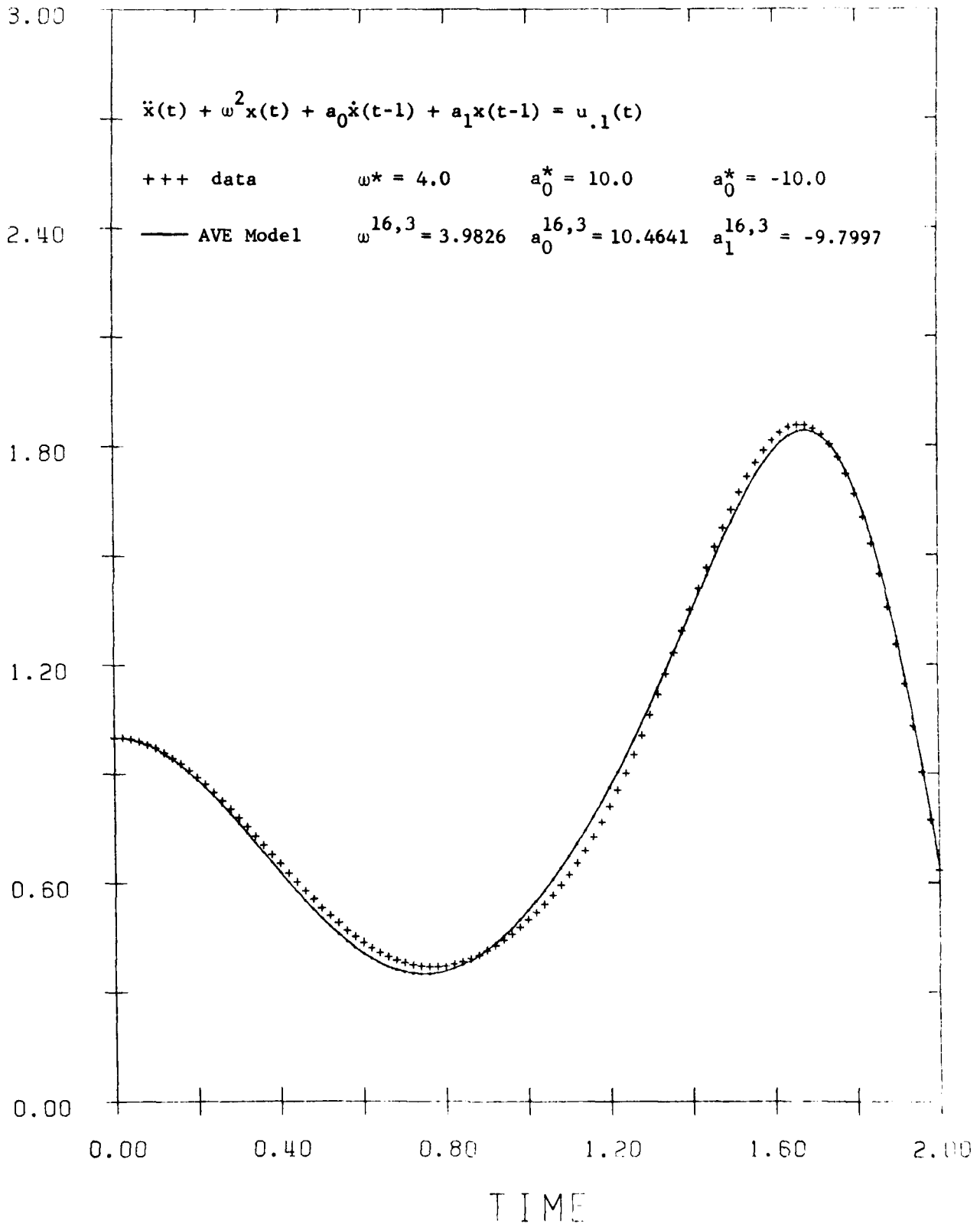
FIGURE 02.1.2

02.1N16S   ITR= 0

$$\ddot{x}(t) + \omega^2 x(t) = a_0 \dot{x}(t-1) + a_1 x(t-1) = u_{.1}(t)$$

+ + +   data $\omega^* = 4.0$ $a_0^* = 10.0$ $a_1^* = -10.0$

_____   SPLINE Model $\omega^{16,0} = \sqrt{20}$ $a_0^{16,0} = 0.0$ $a_1^{16,0} = -9.0$

FIGURE 02.1.3

02.1N16S   ITR= 4

$$\ddot{x}(t) + \omega^2 x(t) + a_0 \dot{x}(t-1) + a_1 x(t-1) = u_{.1}(t)$$

+ + + data $\omega^* = 4.0$ $a_0^* = 10.0$ $a_1^* = -10.0$

—— SPLINE Model $\omega^{16,4} = 3.9943$ $a_0^{16,4} = 9.9920$ $a_1^{16,4} = -9.9812$
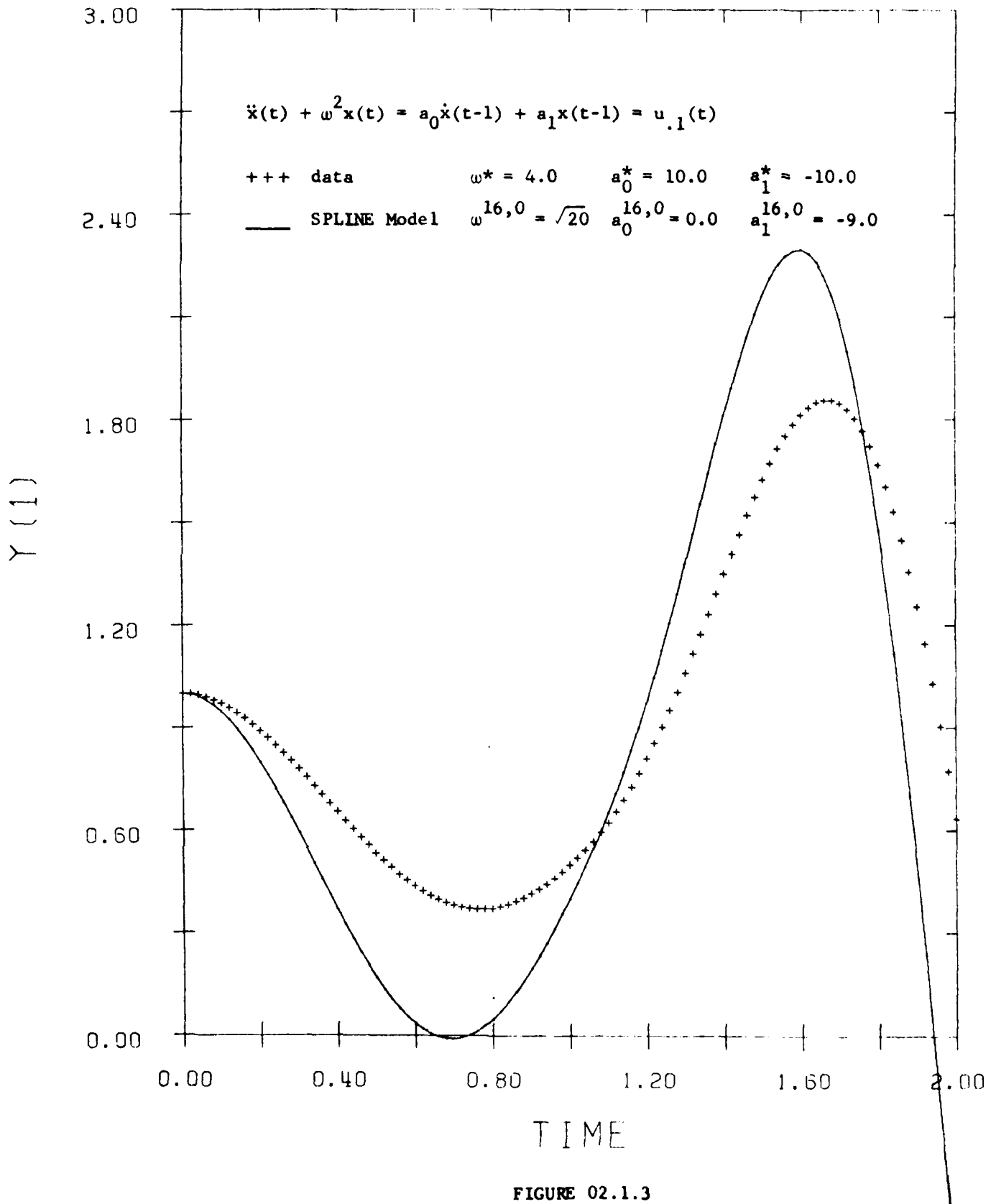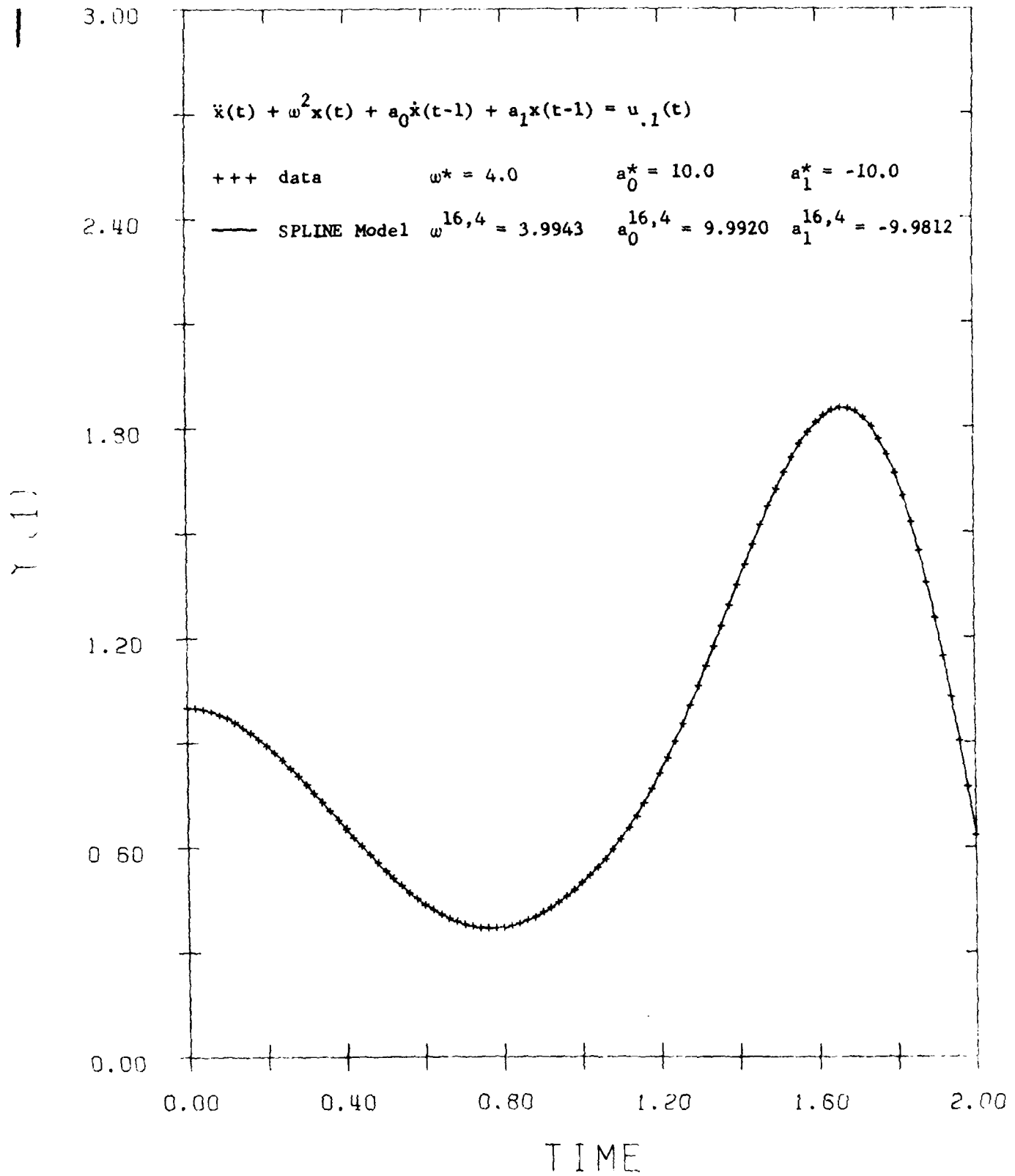
FIGURE 02.1.4

EXAMPLE 02.2

In this problem we seek to estimate the coefficients of the delayed terms and the time delay itself. In particular, we assume that the system is governed by the model

$$\ddot{x}(t) + 16x(t) + a_0\dot{x}(t-r) + a_1x(t-r) = u_{.1}(t) ,$$

with initial data

$$x_0(s) \equiv 1 \; , \; \dot{x}_0(s) \equiv 0 \; , \; -r \leq s \leq 0 \; ,$$

and output

$$y(t) = x(t) ,$$

and the true parameters to be estimated are $a_0^* = 10.0$, $a_1^* = -10.0$ and $r^* = 1.0$. Start-up values for each run were

$$a_0^{N,0} = 11.0 \; , \quad a_1^{N,0} = -9.0 \; , \; r^{N,0} = 1.2 .$$

Convergence results for this example are summarized in Tables 02.2.1 and 02.2.2. At $N = 16$ the relative ℓ_1 error for AVE is approximately 3.5%, while the $N = 16$ SPLINE scheme produced a relative ℓ_1 error of less than 1%.

Figures 02.2.1 and 02.2.2 show the $N = 4$ converged data fits for AVE and SPLINE, respectively. For $N \geq 8$, the data fits are nearly perfect and are not shown.

AVE				
N	$\hat{a}_0^N$	$\hat{a}_1^N$	$\hat{r}^N$	$|e_N|$
2	did not converge			—
4	54.5124	-9.1876	2.4190	46.7439
8	19.4941	-9.4927	1.3506	10.3520
16	10.6433	-9.9089	.9998	.7346
$\gamma^* =$	10.0000	-10.0000	1.0000	

TABLE 02.2.1

SPLINE				
N	$\hat{a}_0^N$	$\hat{a}^N$	$\hat{r}^N$	$|e_N|$
2	9.2585	-10.5360	1.0908	1.3683
4	10.0927	-10.0619	1.0076	.1622
8	9.9724	-10.0177	1.0010	.0463
16	9.9811	-10.0108	1.0017	.0314
$\gamma^* =$	10.0000	-10.0000	1.0000	

TABLE 02.2.2

02.2N4AV   ITR= 4

$$\ddot{x}(t) + 16x(t) + a_0\dot{x}(t-r) + a_1x(t-r) = u_{.1}(t)$$

+ + +   data $a_0^* = 10.0$ $a_1^* = -10.0$ $r^* = 1.0$

———   AVE Model $a_0^{4,4} = 54.5124$ $a_1^{4,4} = -9.1876$ $r^{4,4} = 2.4190$
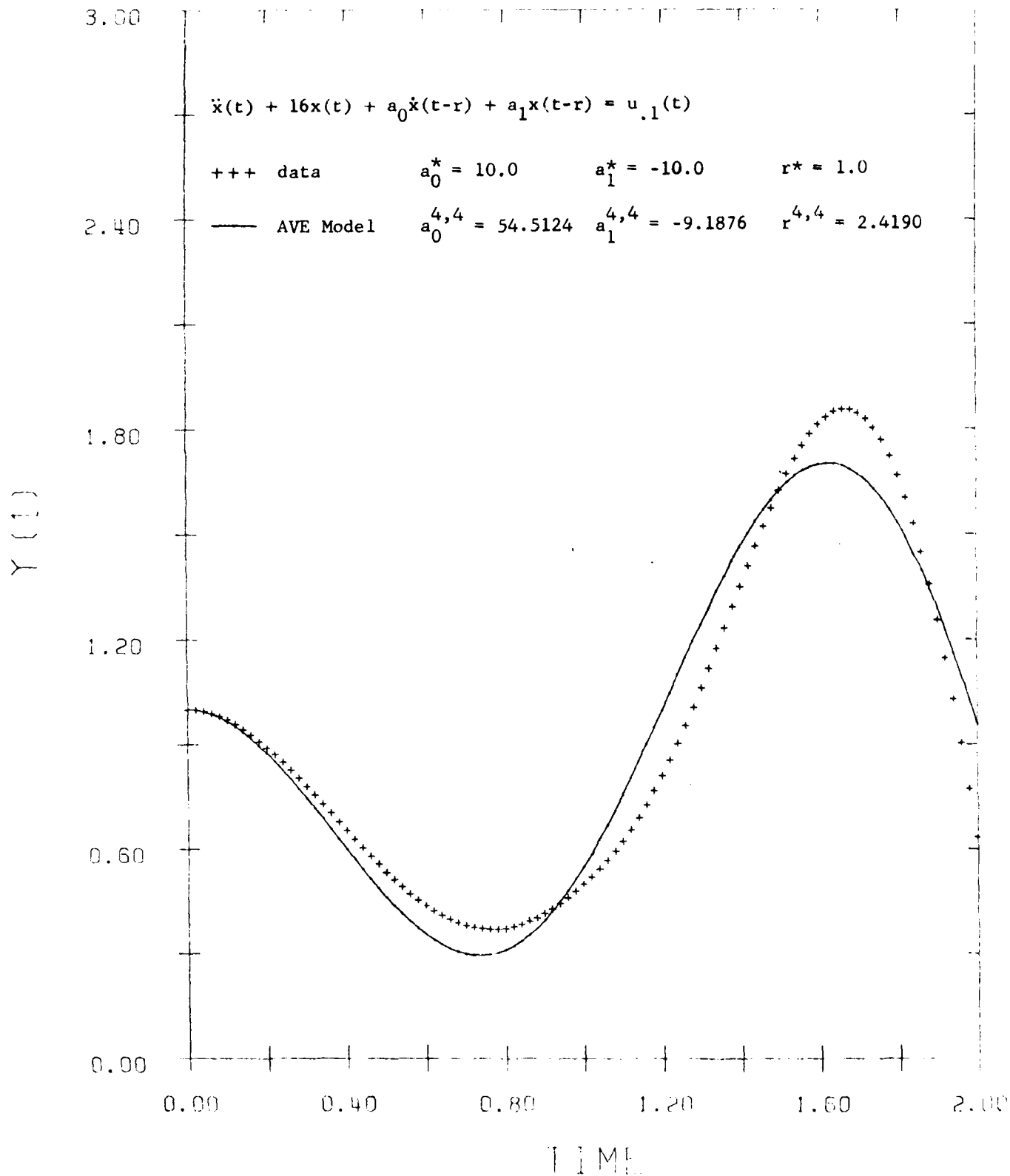
FIGURE 02.2.1

02.2N4SP ITR= 4

$$\ddot{x}(t) + 16x(t) + a_0\dot{x}(t-r) + a_1x(t-r) = u_{.1}(t)$$

+++ data $\qquad a_0^* = 10.0 \qquad a_1^* = -10.0 \qquad r^* = 1.0$

—— SPLINE Model $\qquad a_0^{4,4} = 10.0927 \qquad a_1^{4,4} = -10.0619 \qquad r^{4,4} = 1.0076$
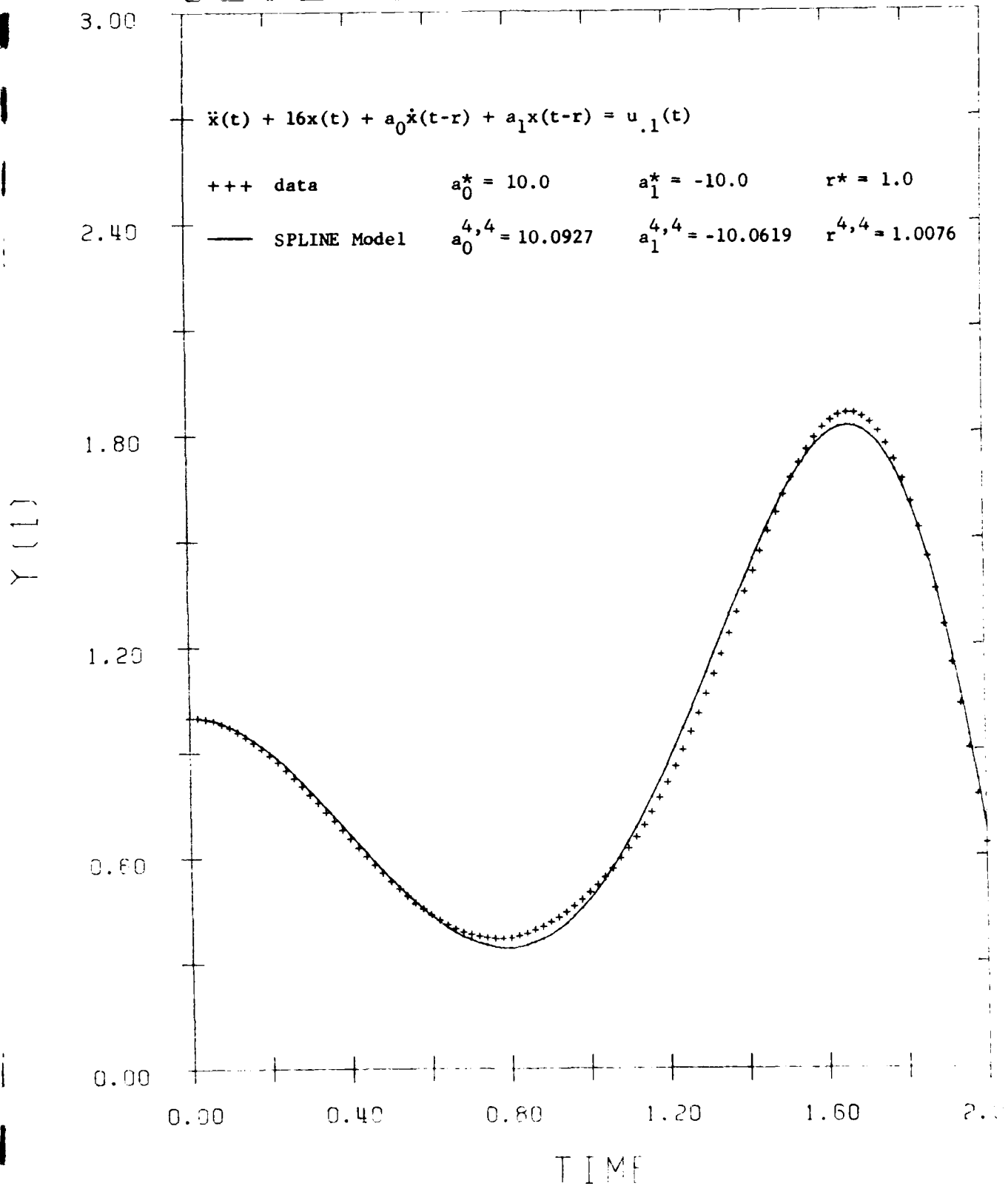
FIGURE 02.2.2

ID MODEL 03

This model is also a harmonic oscillator; however we shall use both position and velocity as output data. In particular, the model is goverred by the equation

$$\ddot{x}(t) + 4x(t) + \dot{x}(t-1) - x(t-1) = u_{.1}(t) ,$$

or in equivalent vector form,

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -4 & 0 \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} x_1(t-1) \\ x_2(t-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_{.1}(t) ,$$

with initial data

$$\begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix}(s)_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} , \quad -1 \le s \le 0 ,$$

and vector output

$$y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} .$$

This system was solved analytically to obtain data at 101 equally spaced points for Examples 03.1 - 03.4.

EXAMPLE 03.1

For this problem we seek to estimate two of the systems co-efficients. In particular, the model is assumed to be governed by the equation

$$\ddot{x}(t) + 4x(t) + a_0\dot{x}(t-1) + a_1x(t-1) = u_{.1}(t)$$

with initial data

$$x_0(s) \equiv 1 , \quad \dot{x}_0(s) \equiv 0 , \quad -1 \leq s \leq 0 ,$$

and vector output (both position and velocity)

$$y(t) = \begin{bmatrix} y_1(t) \\ \\ y_2(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ \\ \dot{x}(t) \end{bmatrix} .$$

The true parameters $a_0^* = 1.0$ and $a_1^* = -1.0$ were estimated using start-up values of

$$a_0^{N,0} = .75 , \quad a_1^{N,0} = -.75 .$$

Runs with $N = 2, 4, 8$ and 16 were made and the convergence results are summarized in Tables 03.1.1 - 03.1.2. Note that at $N = 16$, the AVE scheme produced parameter estimates considerably worse (about 16% "relative ℓ_1 error") than the $N = 8$ estimate (about 7% error). The $N = 16$ SPLINE procedure gave estimates with less than 2% relative error.

Typical data fits are illustrated in Figures 03.1.1 - 03.1.8.
Figures 03.1.1 - 03.1.2 are the N = 8 AVE start-up data fits for
the position (Y(1)) and velocity (Y(2)), respectively. The con-
verged N = 8 AVE data fits are shown in Figures 03.1.2 - 03.1.4.
Figures 03.1.5 - 03.1.8 show the same data fits for the N = 8
SPLINE procedure.

AVE			
N	$\hat{a}_0^N$	$\hat{a}_1^N$	$\lvert e_N \rvert$
2	1.1437	- .8789	.2648
4	1.1504	- .9221	.2283
8	1.0951	- .9579	.1372
16	.7215	-1.0483	.3261
$\gamma^* =$	1.0000	-1.0000	

TABLE 03.1.1

	SPLINE		
N	$\hat{a}_0^N$	$\hat{a}_1^N$	$|e_N|$
2	1.2474	-1.0991	.3465
4	1.0256	-1.0350	.0606
8	.9936	-1.0137	.0398
16	.9739	-1.0100	.0361
$\gamma^* =$	1.0000	-1.0000	

TABLE 03.1.2

03.1N8AV   ITR= 0

$$\ddot{x}(t) + 4x(t) + a_0\dot{x}(t-1) + a_1x(t-1) = u_{.1}(t)$$

$+++$ data $a_0^* = 1.0$ $a_1^* = -1.0$

——— AVE Model $a_0^{8,0} = .75$ $a_1^{8,0} = -.75$

Y_1

1.50

1.10

0.70

0.30

-0.11

-0.50

0.00   0.40   0.80   1.20   1.60   2.00

TIME
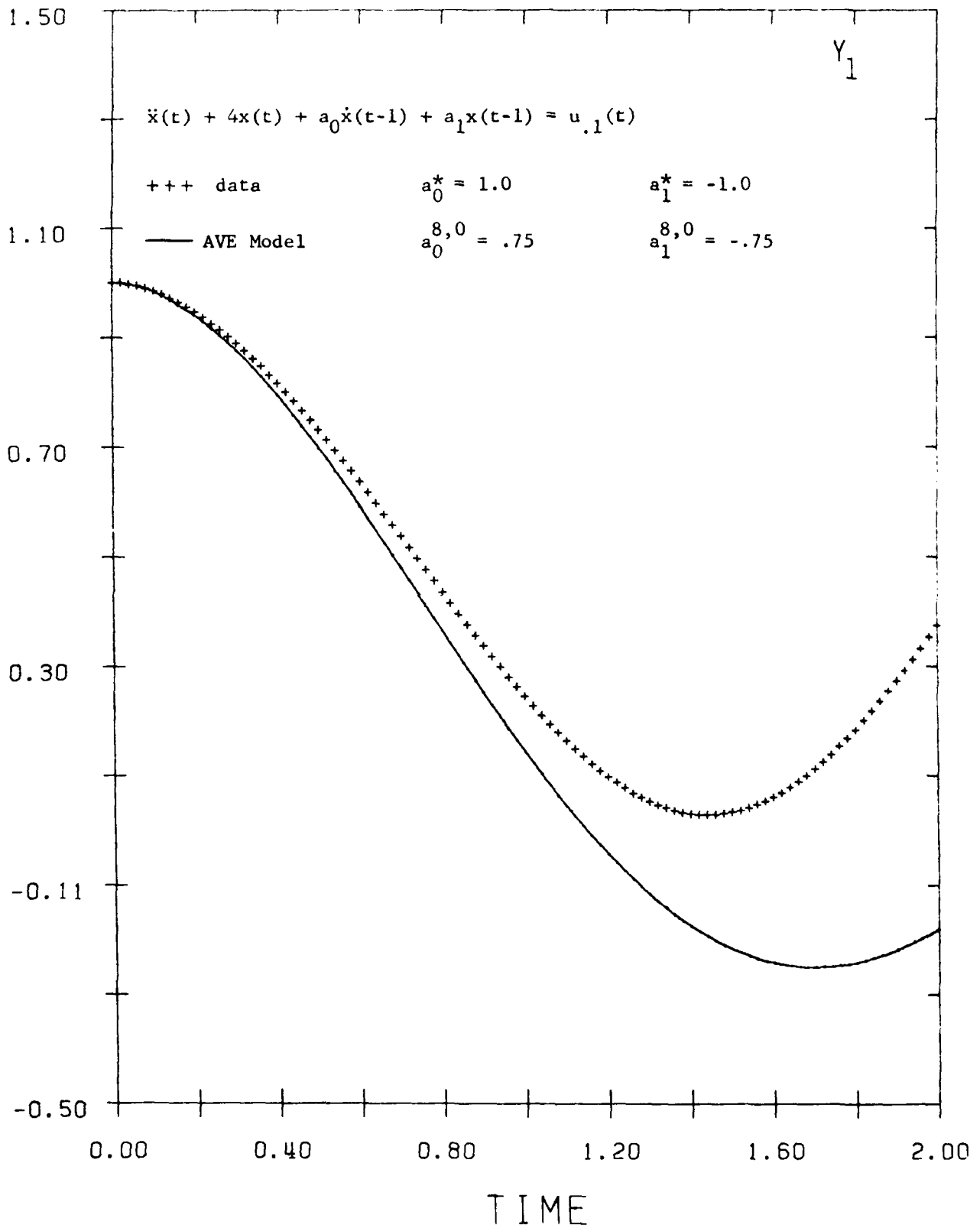
FIGURE 03.1.1

03.1N8AV ITR= 0

$$\ddot{x}(t) + 4x(t) + a_0 x(t-1) + a_1 x(t-1) = u_{.1}(t)$$

+++ data $\quad a_0^* = 1.0 \qquad a_1^* = -1.0$

—— AVE Model $\quad a_0^{8,0} = .75 \qquad a_1^{8,0} = -.75$

Y_2

TIME

FIGURE 03.1.2

Ø3.1N8AV   ITR= 10

Y_1

$\ddot{x}(t) + 4x(t) + a_0\dot{x}(t-1) + a_1x(t-1) = u_{.1}(t)$

+ + +   data $a_0^* = 1.0$ $a_1^* = -1.0$

———   AVE Model $a_0^{8,10} = 1.0951$ $a_1^{8,10} = -.9579$

TIME
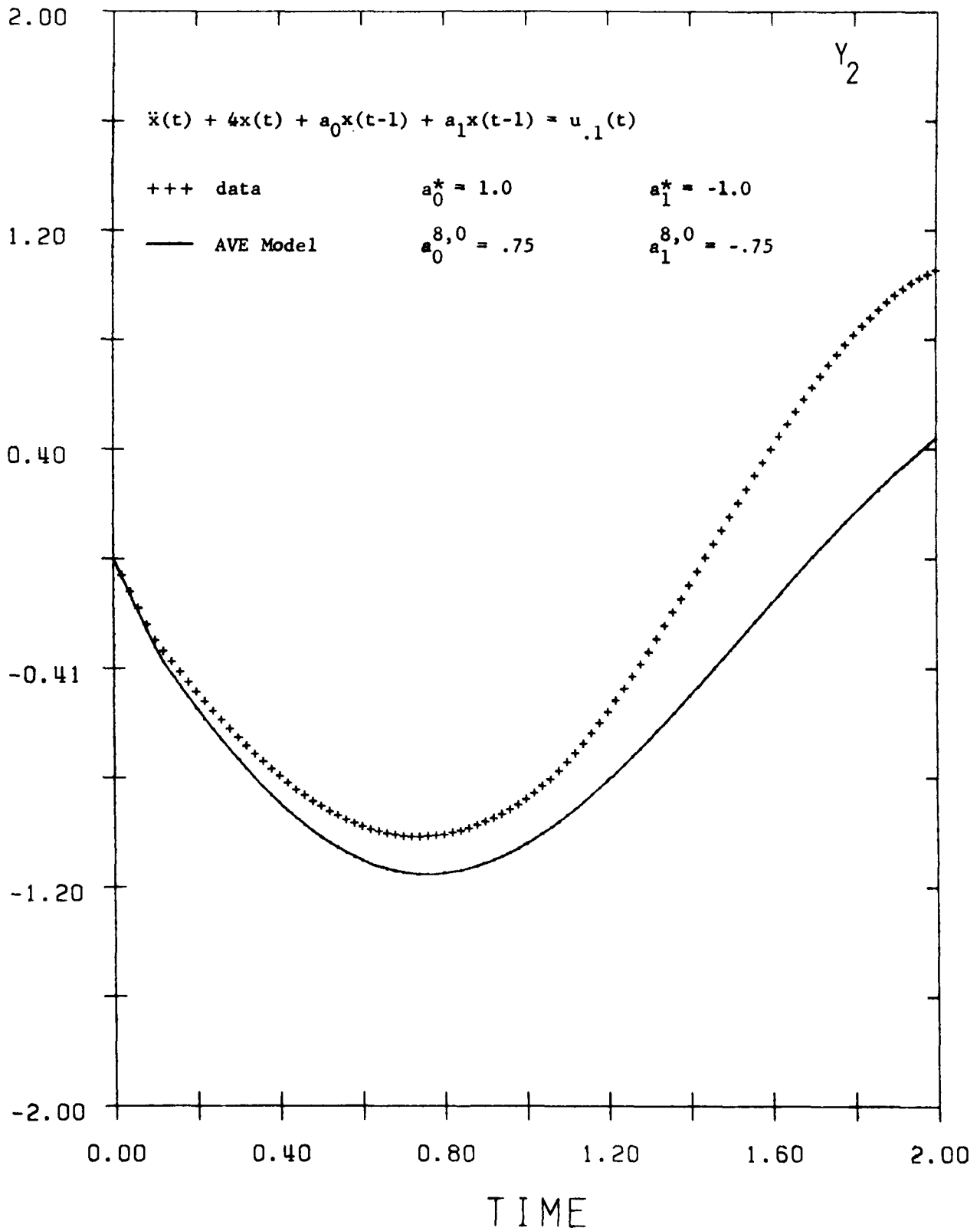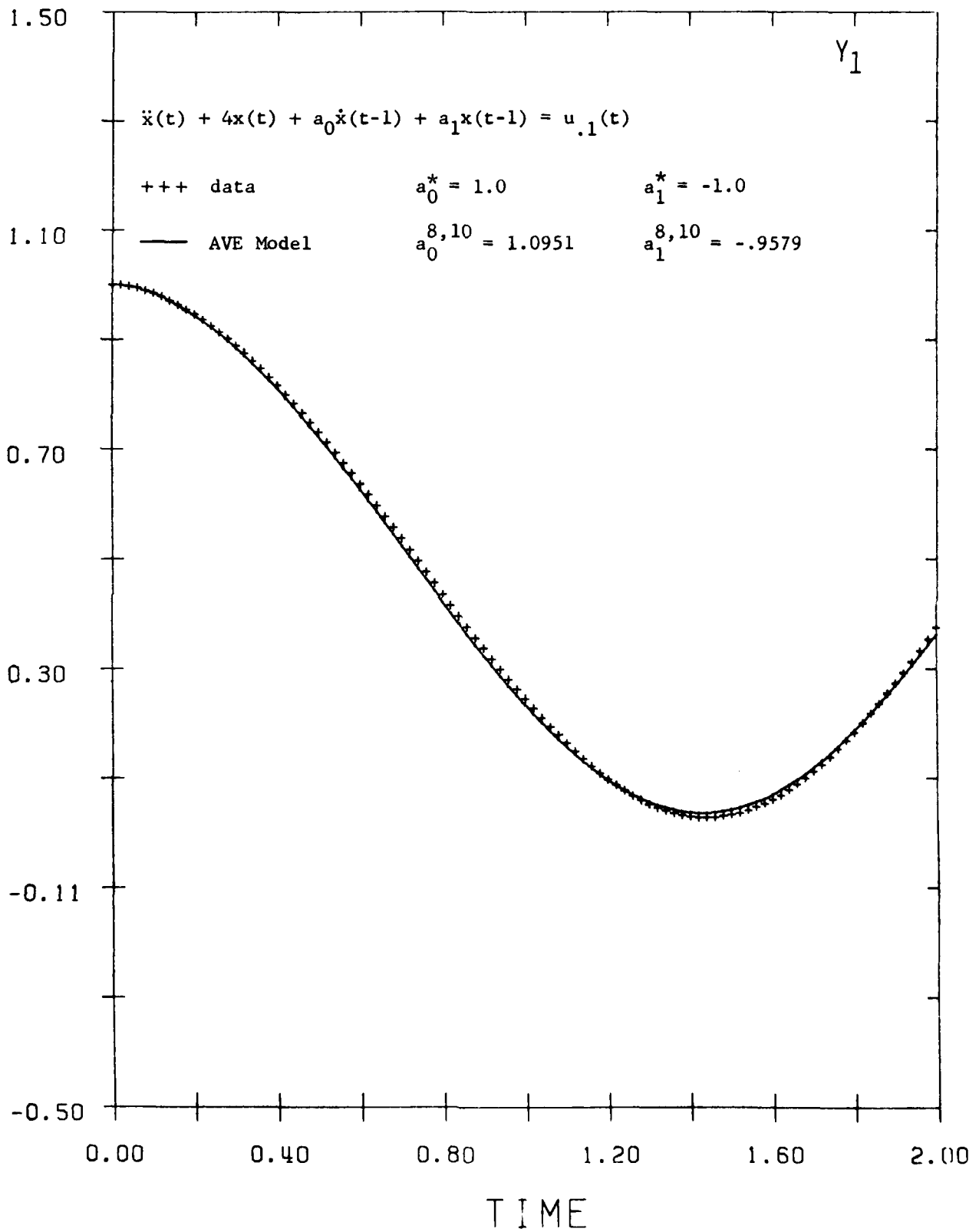
FIGURE 03.1.3

03.1N8AV   ITR= 10

Y_2

$$\ddot{x}(t) + 4x(t) + a_0\dot{x}(t-1) + a_1x(t-1) = u_{.1}(t)$$

+ + +   data $a_0^* = 1.0$ $a_1^* = -1.0$

——   AVE Model $a_0^{8,10} = 1.0951$ $a_1^{8,10} = -.9579$

TIME
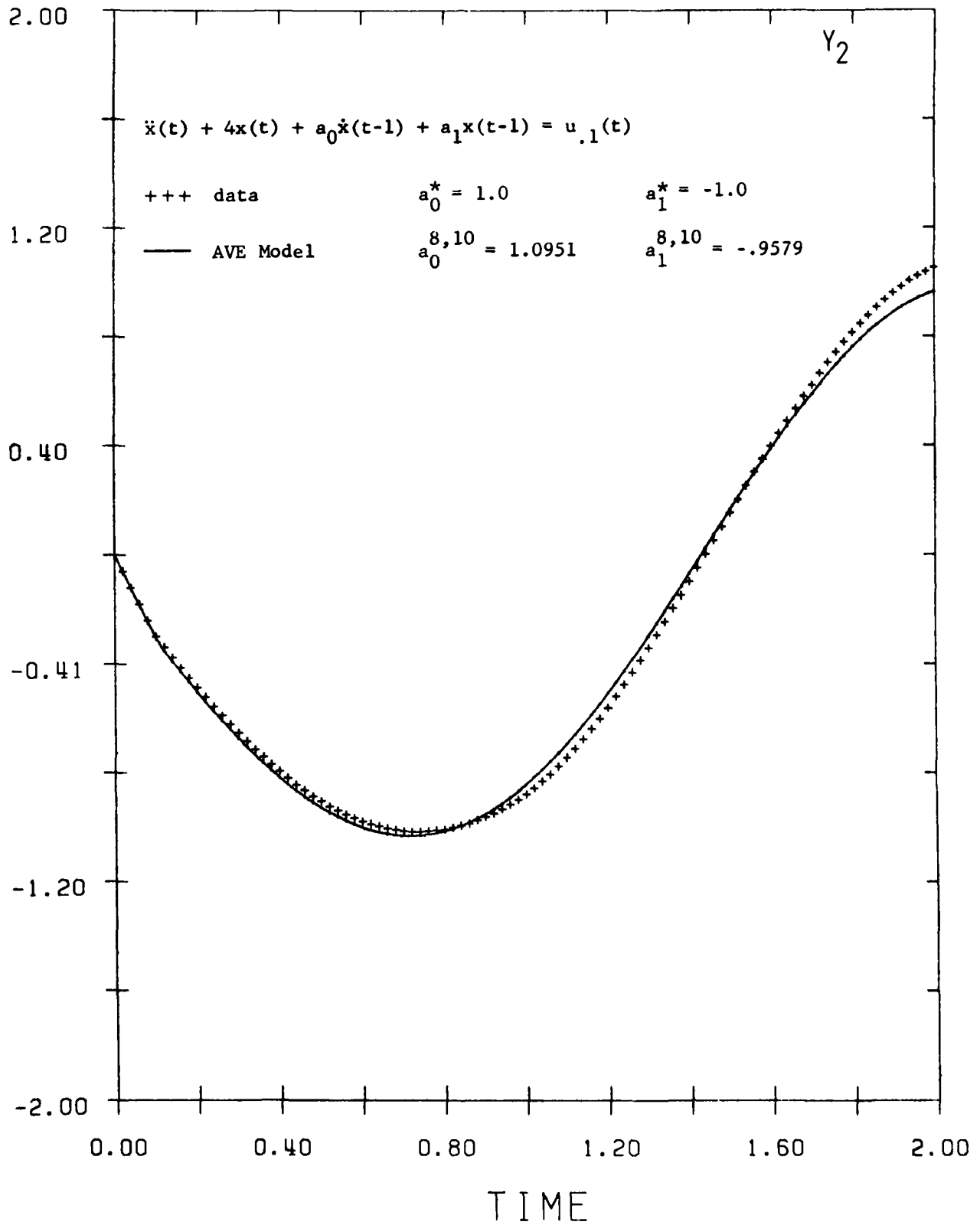
FIGURE 03.1.4

03.1N8SP   ITR= 0

Y_1

$$\ddot{x}(t) + 4x(t) + a_0\dot{x}(t-1) + a_1x(t-1) = u_{.1}(t)$$

+ + +   data $a_0^* = 1.0$ $a_1^* = -1.0$

———   SPLINE Model $a_0^{8,10} = .75$ $a_1^{8,10} = -.75$

TIME
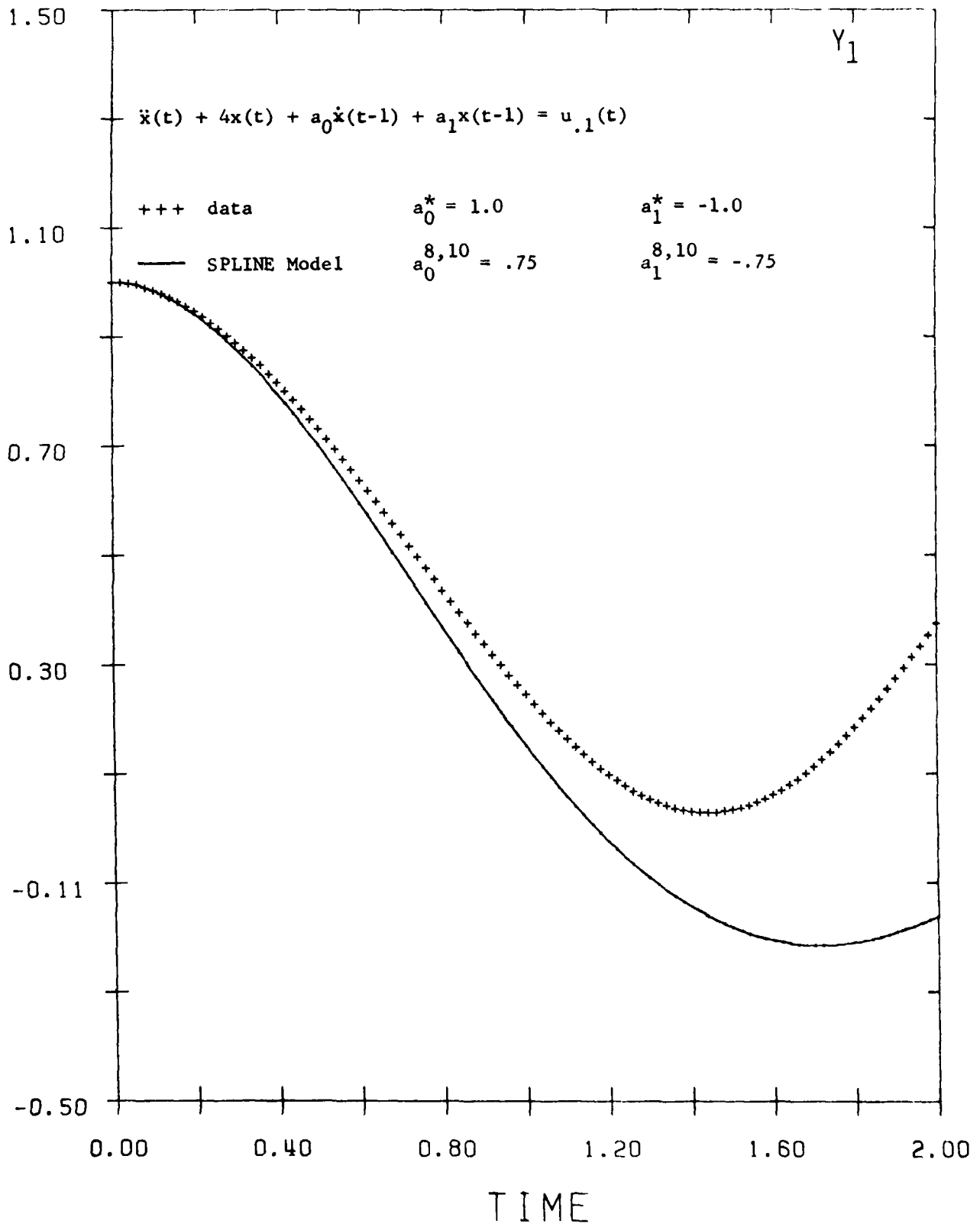
FIGURE 03.1.5

03.1N8SP   ITR= 0

Y_2

$\ddot{x}(t) + 4x(t) + a_0\dot{x}(t-1) + a_1x(t-1) = u_{.1}(t)$

+ + + data $a_0^* = 1.0$ $a_1^* = -1.0$

——— SPLINE Model $a_0^{8,0} = .75$ $a_0^{8,0} = -.75$

TIME
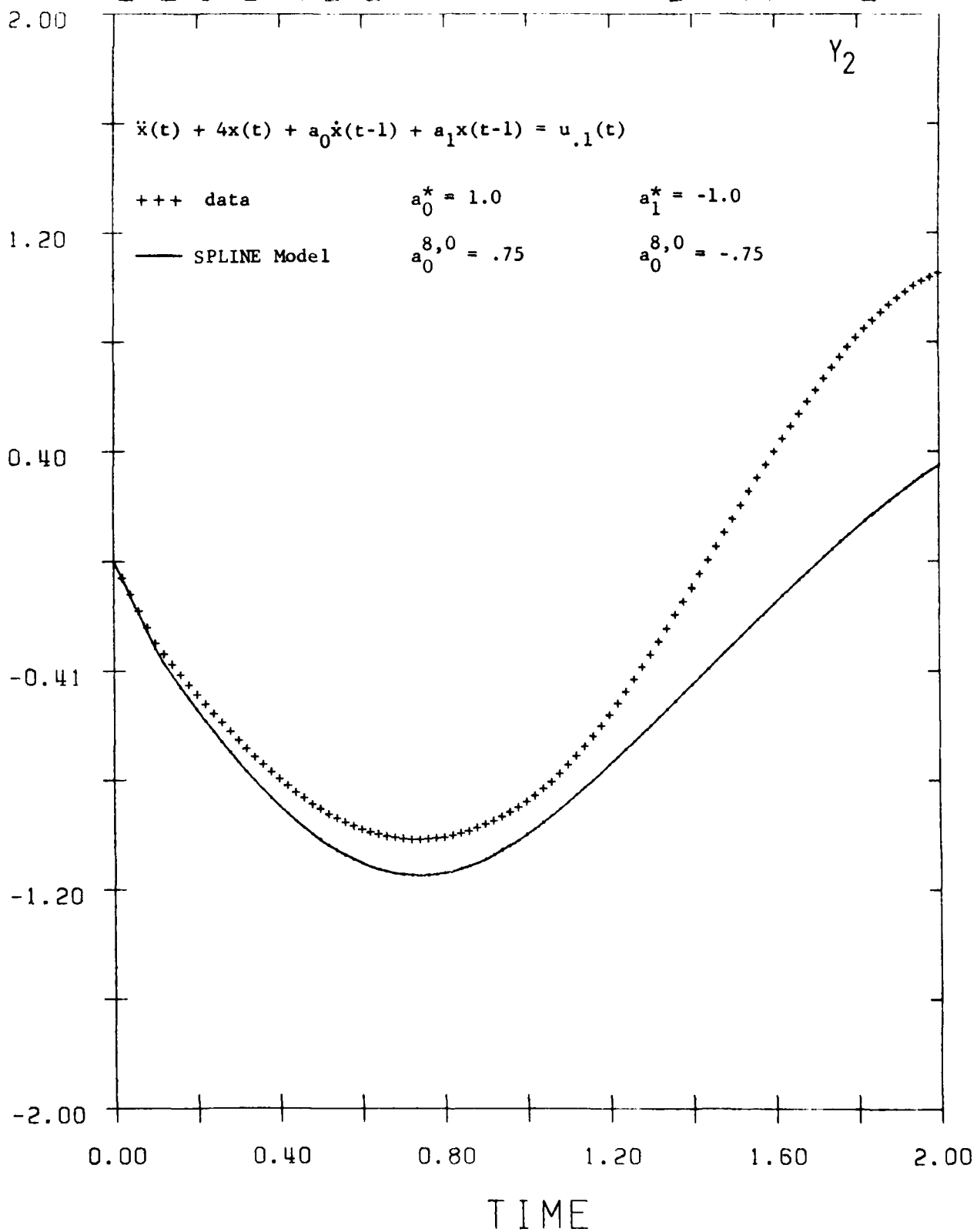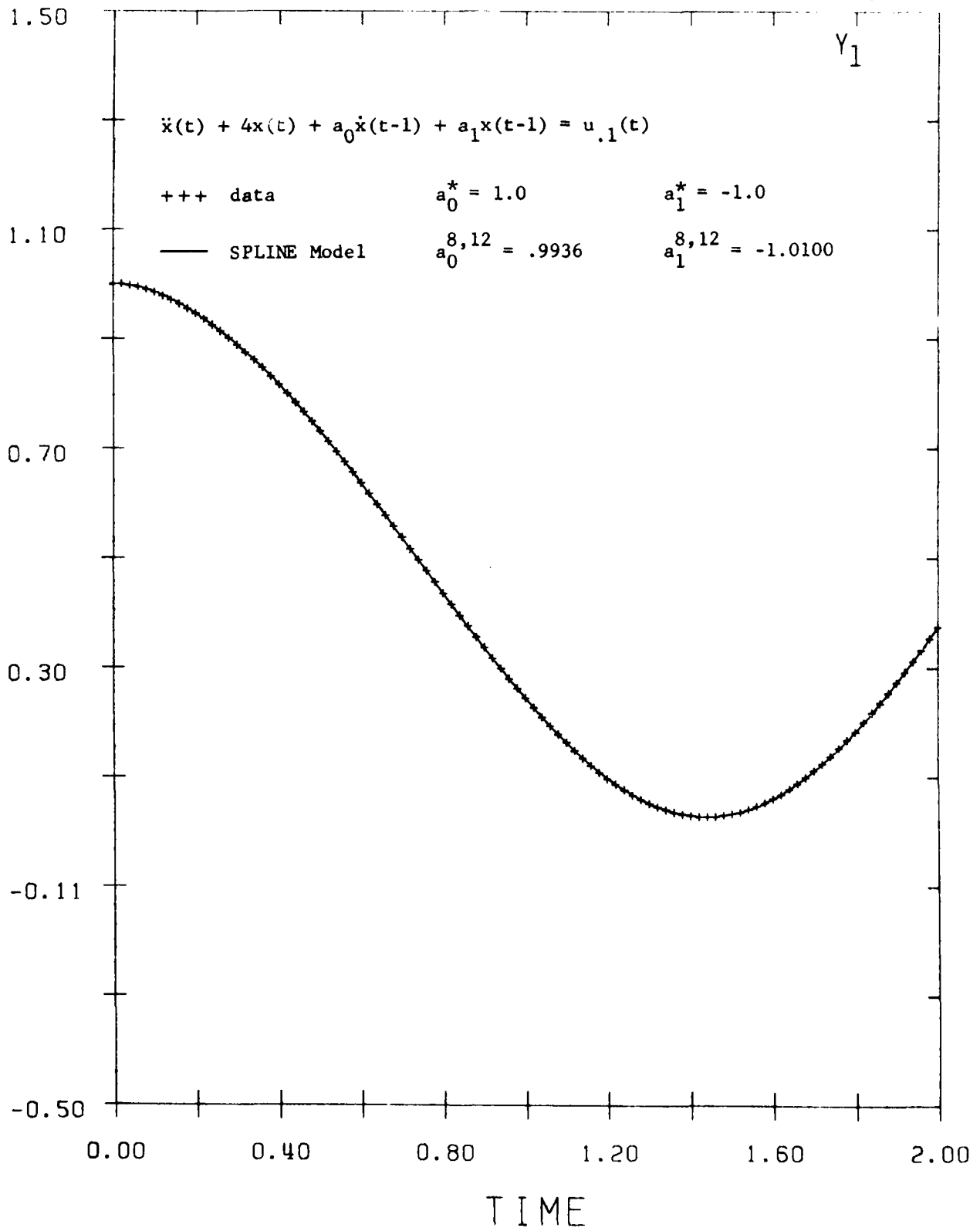
FIGURE 03.1.6
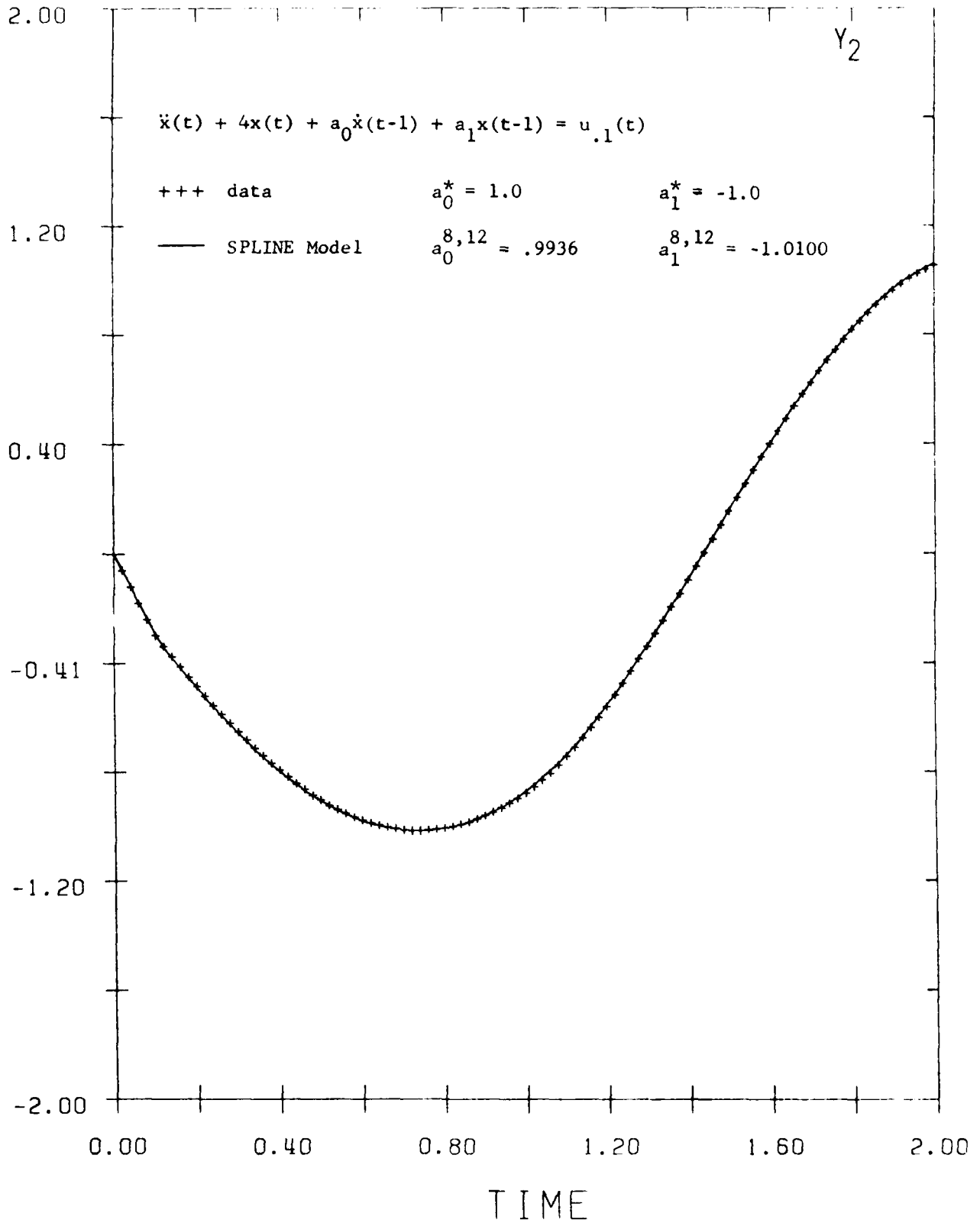
FIGURE 03.1.7

FIGURE 03.1.8

EXAMPLE 03.2

In this example we attempt to estimate all of the system co-efficients. Consequently, the model is assumed to be of the form

$$\ddot{x}(t) + \omega^2 x(t) + a_0 \dot{x}(t-1) + a_1 x(t-1) = u_{.1}(t)$$

with initial data

$$x_0(s) \quad 1 \quad , \quad \dot{x}_0(s) = 0$$

and output

$$y(t) = \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} \quad .$$

The true parameters $\omega^* = 2.0$, $a_0^* = 1.0$, $a_1^* = -1.0$ were estimated using start-up values of

$$\omega^{N,0} = \sqrt{3.0} \quad , \quad a_0^{N,0} = .75 \quad , \quad a_1^{N,0} = -.75 \quad .$$

This example is again typical in that the SPLINE scheme pro-duced better estimates than the AVE scheme (although the N = 2 SPLINE run did not converge). The convergence of the parameter estimates is summarized in Tables 03.2.1 - 03.2.2. The N = 16 results show that the relative ℓ_1 error is about 5% for AVE and 1% for SPLINE. The data fits for N = 8 and 16 were nearly perfect fits for both AVE and SPLINE. Consequently, no plots are given.

AVE				
N	$\hat{x}^N$	$\hat{a}_0^N$	$\hat{a}_1^N$	$|e_N|$
2	1.2398	3.8802	1.3463	5.9867
4	1.7711	2.1018	- .2201	2.1106
8	1.8955	1.5372	- .6522	.9895
16	1.9404	.9505	- .9033	.2050
$\gamma^* =$	2.0000	1.0000	-1.0000	

TABLE 03.2.1

SPLINE				
N	$\hat{x}^N$	$\hat{a}_0^N$	$\hat{a}_1^N$	$|e_N|$
2	d i d	n o t	c o n v e r g e	——
4	2.0320	.9324	-1.1136	.2132
8	1.9995	.9956	-1.0124	.0173
16	1.9903	1.0149	- .9840	.0406
γ^*	2.0000	1.0000	1.0000	

TABLE 03.2.2

EXAMPLE O3.4

For this example we seek to estimate an initial function, two system coefficients and the time delay. In particular we assume that the initial position is a constant but unknown value and hence the model takes the form

$$\ddot{x}(t) + 4x(t) + a_0\dot{x}(t-r) + a_1 x(t-r) = u_{.1}(t)$$

with (partially unknown) initial data

$$x_0(s) = C \ , \ \dot{x}_0(s) = 0 \ , \ -r \leq s \leq 0$$

and vector output

$$y(t) = \begin{bmatrix} x(t) \\ \\ \dot{x}(t) \end{bmatrix} \ .$$

The parameters $C^* = 1.0$, $a_0^* = 1.0$, $a_1^* = -1.0$ and $r^* = 1.0$ were estimated using start-up values

$$C^{N,0} = 0.0, \ a_0^{N,0} = .9, \ a_1^{N,0} = -.9, \ r^{N,0} = .9 \ .$$

For each N = 2, 4, 8 and 16, the AVE scheme did not converge. At N = 2 the SPLINE scheme did not converge; however, at N = 4, 8 and 16 the SPLINE procedure converged to good estimates of the parameters. This example was also run with other start-ups and it was observed that unless the start-ups were reasonably close
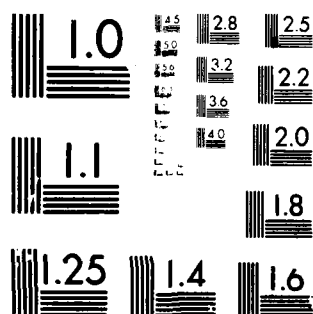
MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

(i.e. as above) the SPLINE scheme also diverged. The parameter estimates produced by the SPLINE scheme are listed in Table 03.4.1. The error $|e_N|$ listed in this table is the "ℓ_1 error"

$$|e_N| = \|\varphi_1^* - \hat{\varphi}_1^N\|_{L_2} + |r^* - \hat{r}^N| + |a_0^* - \hat{a}_0^N| + |a_1^* - \hat{a}_0^N|$$

$$= |C^* - \hat{C}^N| + |r^* - \hat{r}^N| + |a_0^* - \hat{a}_0^N| + |a_1^* - \hat{a}_1^N| .$$

Observe that the relative error at each $N = 4, 8, 16$ is between 1 and 2 percent.

Figures 03.4.1-03.4.2 show the $N = 4$ SPLINE data fits. Note that the MLE required 29 iterations to converge.

		SPLINE			
N	$\hat{C}^N$	$\hat{r}^N$	$\hat{a}_0^N$	$\hat{a}_1^N$	$|e_N|$
2	d i d	n o t	c o n v e r g e		——
4	.9960	.9440	1.0135	-1.0085	.0820
8	.9998	.9999	.9937	-1.0137	.0203
16	1.0011	1.0156	.9874	-1.0131	.0434
$\gamma^* =$ 1.0000	1.0000	1.0000	-1.0000		

TABLE 03.4.1

03.4N4SP   ITR= 29

$\ddot{x}(t) + 4x(t) + a_0\dot{x}(t-r) + a_1 x(t-r) = u_{.1}(t)$

$x_0(s) \equiv C , \dot{x}_0(s) \equiv 0$

+++ data   C* = 1.0   r* = 1.0 $a_0^* = 1.0$ $a_1^* = -1.0$

____ SPLINE Model $c^{4,29} = .9960$ $r^{4,29} = .9440$ $a_0^{4,29} = 1.0135$ $a_1^{4,29} = -1.0065$

TIMF

FIGURE 03.4.1

03.4N4SP   ITR 29

$$\ddot{x}(t) + 4x(t) + a_0\dot{x}(t-r) + a_1 x(t-r) = u_{.1}(t)$$

$$x_0(s) \equiv C , \quad \dot{x}_0(s) \equiv 0$$

+++ data   C* = 1.0   r* = 1.0 a_0^* = 1.0 a_1^* = -1.0

——SPLINE Model $C^{4,29}$=.9960 $r^{4,29}$=.9440 $a_0^{4,29}$=1.0135 $a_1^{4,29}$=-1.0085

TIME
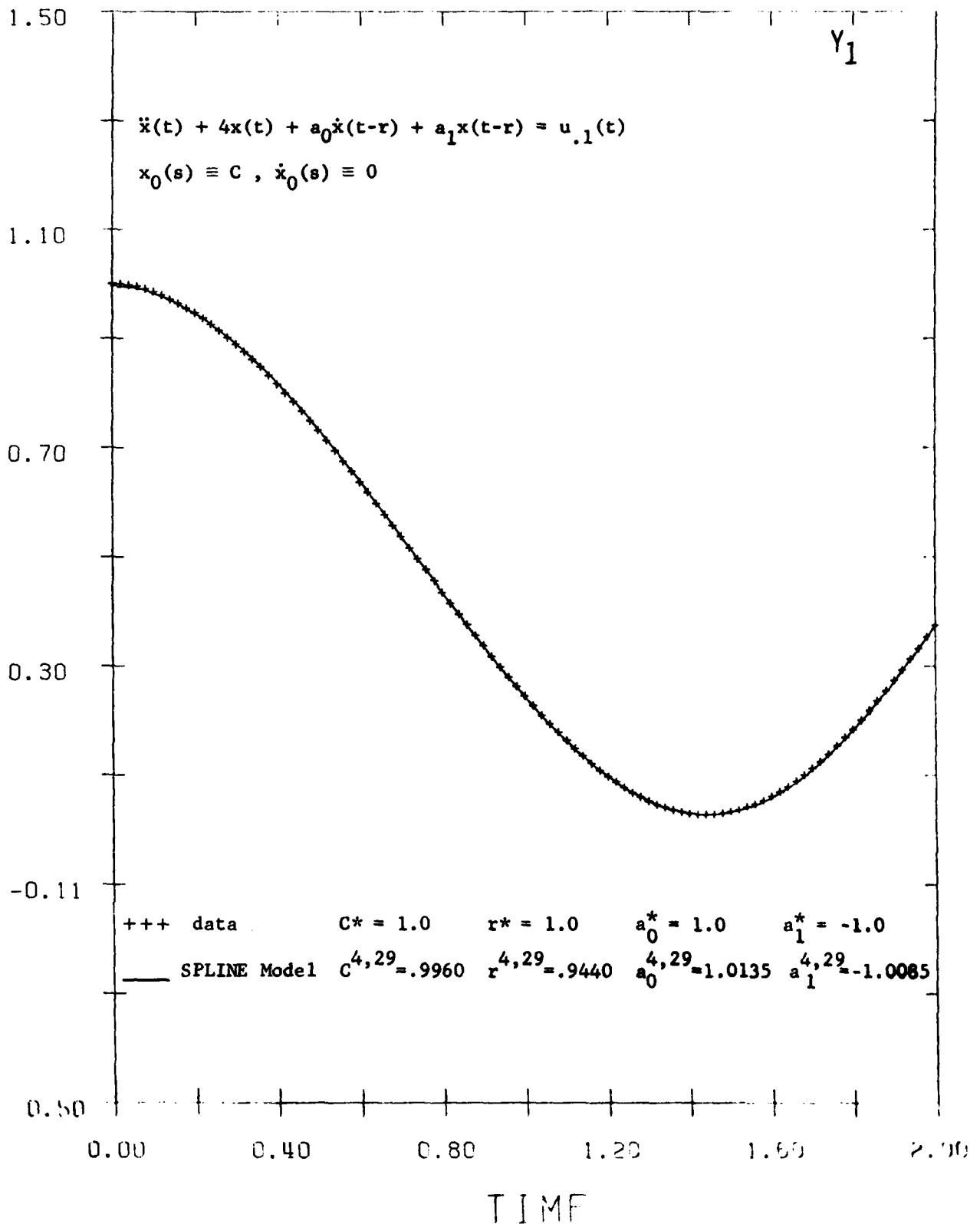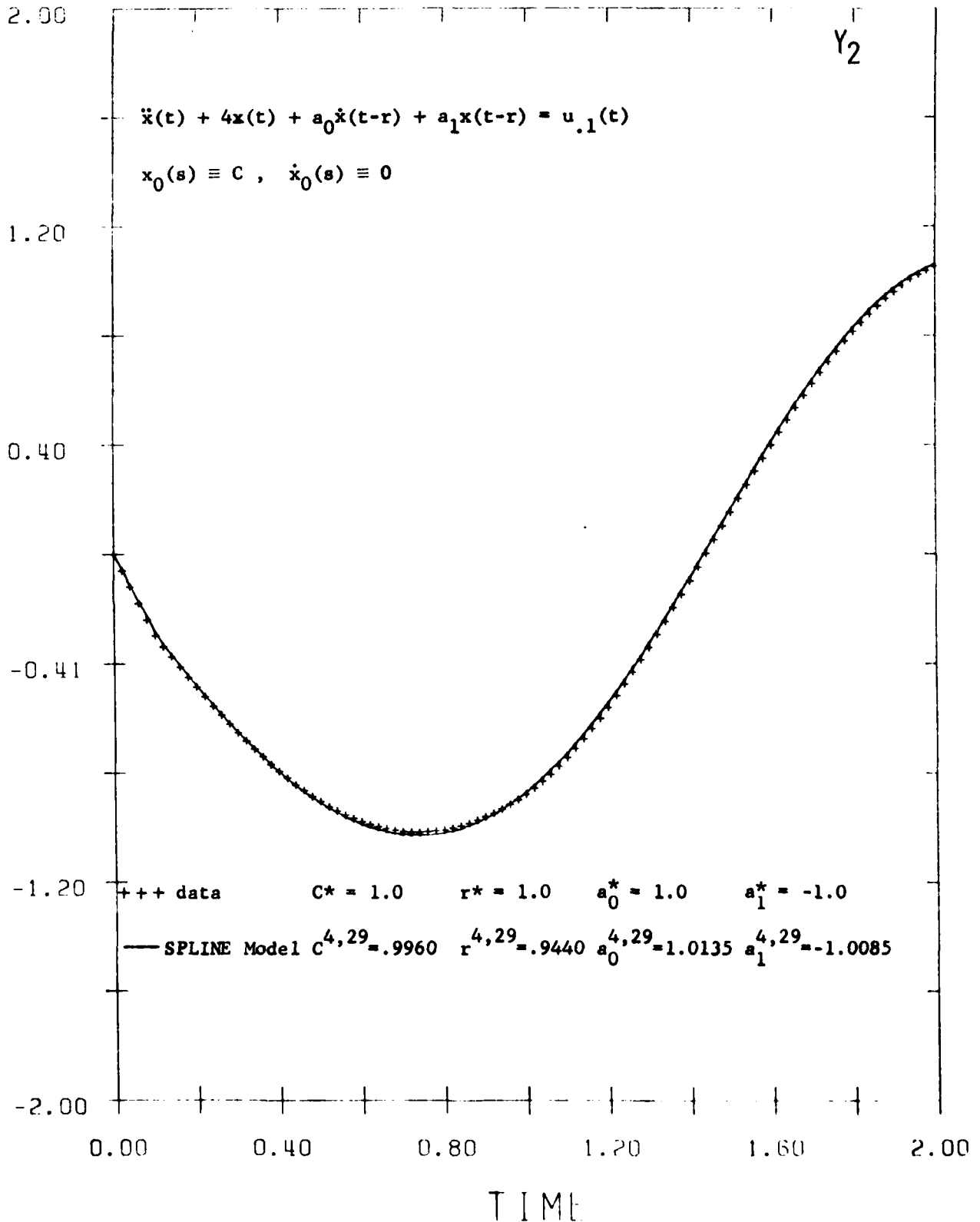
FIGURE 03.4.2

ID MODEL 04

This model is the oscillator considered in ID MODEL 03, with noise added to the output. In particular, the system is governed by the equation

$$\ddot{x}(t) + 4x(t) + \dot{x}(t-1) - x(t-1) = u_{.1}(t) ,$$

with initial data

$$x_0(s) \equiv 1 , \quad \dot{x}_0(s) \equiv 0 , \quad -1 \leq s \leq 0 ,$$

and (noisy) output

$$y(t) = \begin{bmatrix} x(t) \\ \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} v_1(t) \\ \\ v_2(t) \end{bmatrix} ,$$

where $v_i(t)$ $(i = 1,2)$ is a computer simulated normal random variable with zero mean and standard deviations of 0.1 on the position data (v_1) and 0.2 on the velocity data (v_2).

The random variables $v_i(t)$ $(i = 1,2)$ were generated using routine GGNQF of the IMSL library (see IMSL Users Guide) and added to the analytic solution $x(t)$ and $\dot{x}(t)$ of the delay equation to produce data $(\bar{y}_i, i = 1,2,\cdots, 101)$ at 101 equally spaced times on $[0,2]$. These values produced rather noisy data, which was used in following examples; 04.2 - 04.3.

EXAMPLE 04.2

For this example we estimate two system coefficients and the time delay. The model is assumed to have the form

$$\ddot{x}(t) + 4x(t) + a_0\dot{x}(t-r) + a_1x(t-r) = u_{.1}(t) \quad ,$$

with initial data

$$x_0(s) \equiv 1 \ , \ \dot{x}_0(s) \equiv 0 \ , \ -r \leq s \leq 0 \ ,$$

and output

$$y(t) = \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} \quad .$$

For N = 2, 4, 8 and 16 the approximating identification problem was formulated and a version of the MLE algorithm described in [9] was used to estimate the parameters $a_0^* = 1.0$, $a_1^* = -1.0$ and $r^* = 1.0$. Start-up values for each run were set at

$$a_0^{N,0} = .75 \ , \ a_1^{N,0} = -.75 \ , \ r^{N,0} = .8 \ .$$

Except for the N = 2 AVE run, each run converged to reasonable estimates for the parameters. Again the SPLINE scheme produced better results. The N = 16 AVE estimates give about 14% relative ℓ_1 error, while the N = 16 SPLINE estimates have about 1% relative ℓ_1 error. The convergence results for this problem are summarized in Tables 04.2.1 and 04.2.2.

Typical data fits for this problem are illustrated in Figures 04.2.1 - 04.2.4. Note that the data produced by adding the simulated noise is indeed very "noisy". However, at N = 8 both AVE and SPLINE do a "good" job of fitting the data. In fact, the N = 8 converged data fits provide nearly perfect matches to the system outputs without the noise.

AVE				
N	$\hat{a}_0^N$	$\hat{a}_1^N$	$\hat{r}^N$	$|e_N|$
2	773.3040	-.9699	65.6704	837.0045
4	2.2051	-.9832	1.8078	2.0297
8	1.3959	-.9947	1.2547	.6559
16	.7740	-1.0664	1.1142	.4066
$\gamma^* =$	1.0000	-1.0000	1.0000	

TABLE 04.2.1

SPLINE				
N	$\hat{a}_0^N$	$\hat{a}_1^N$	$\hat{r}^N$	$|e_N|$
2	1.0813	-1.0254	.8233	.2834
4	1.0001	-1.0283	.9533	.0751
8	.9881	-1.0195	.9937	.0377
16	.9850	-1.0156	1.0013	.0319
$\gamma^* =$	1.0000	-1.0000	1.0000	

TABLE 04.2.2
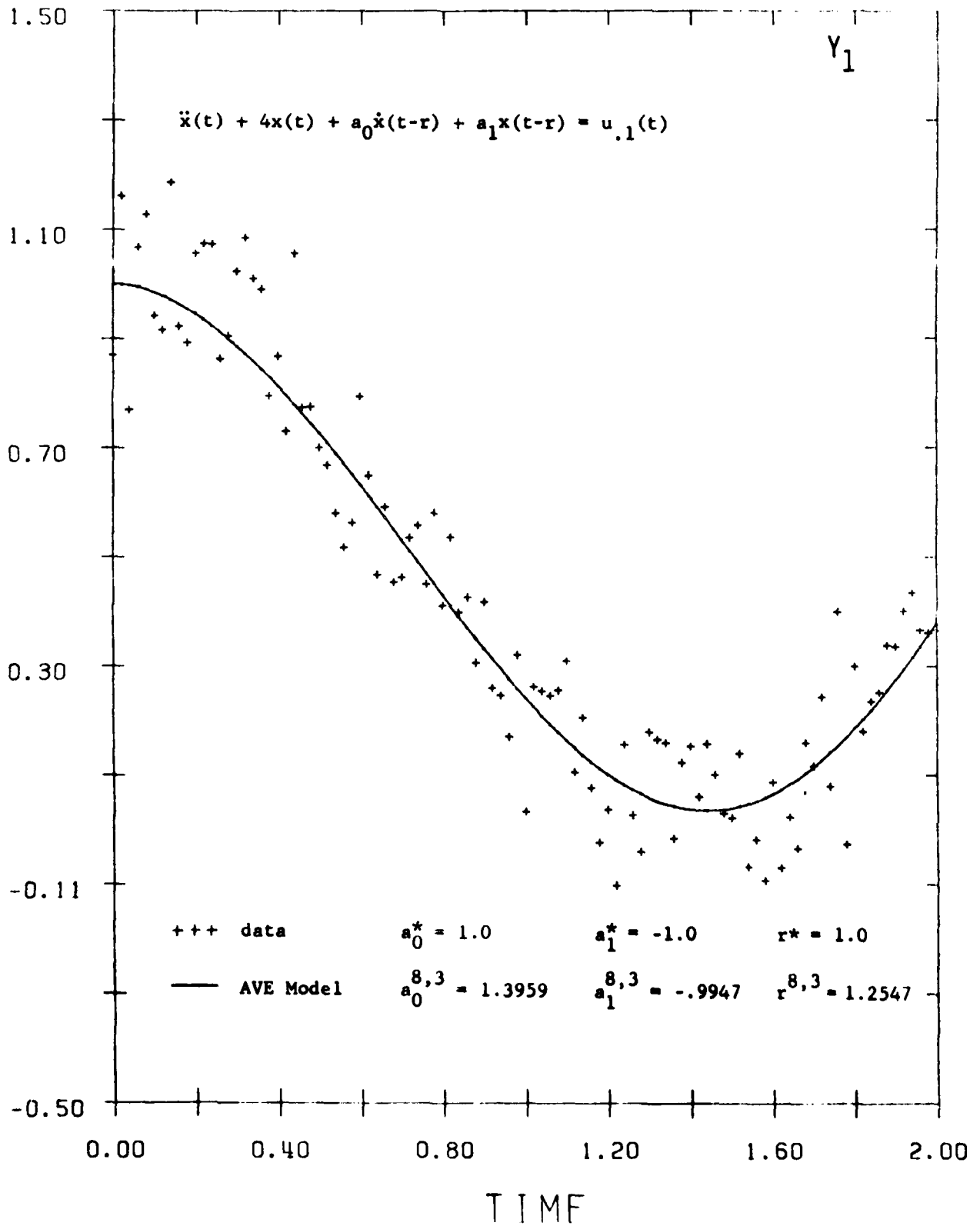
FIGURE 04.2.1

04.2N8AV   ITR= 3

Y_2

$$\ddot{x}(t) + 4x(t) + a_0\dot{x}(t-r) + a_1x(t-r) = u_{.1}(t)$$

+ + + data $a_0^* = 1.0$ $a_1^* = -1.0$ $r^* = 1.0$

—— AVE Model $a_0^{8,3} = 1.3959$ $a_1^{8,3} = -.9947$ $r^{8,3} = 1.2547$
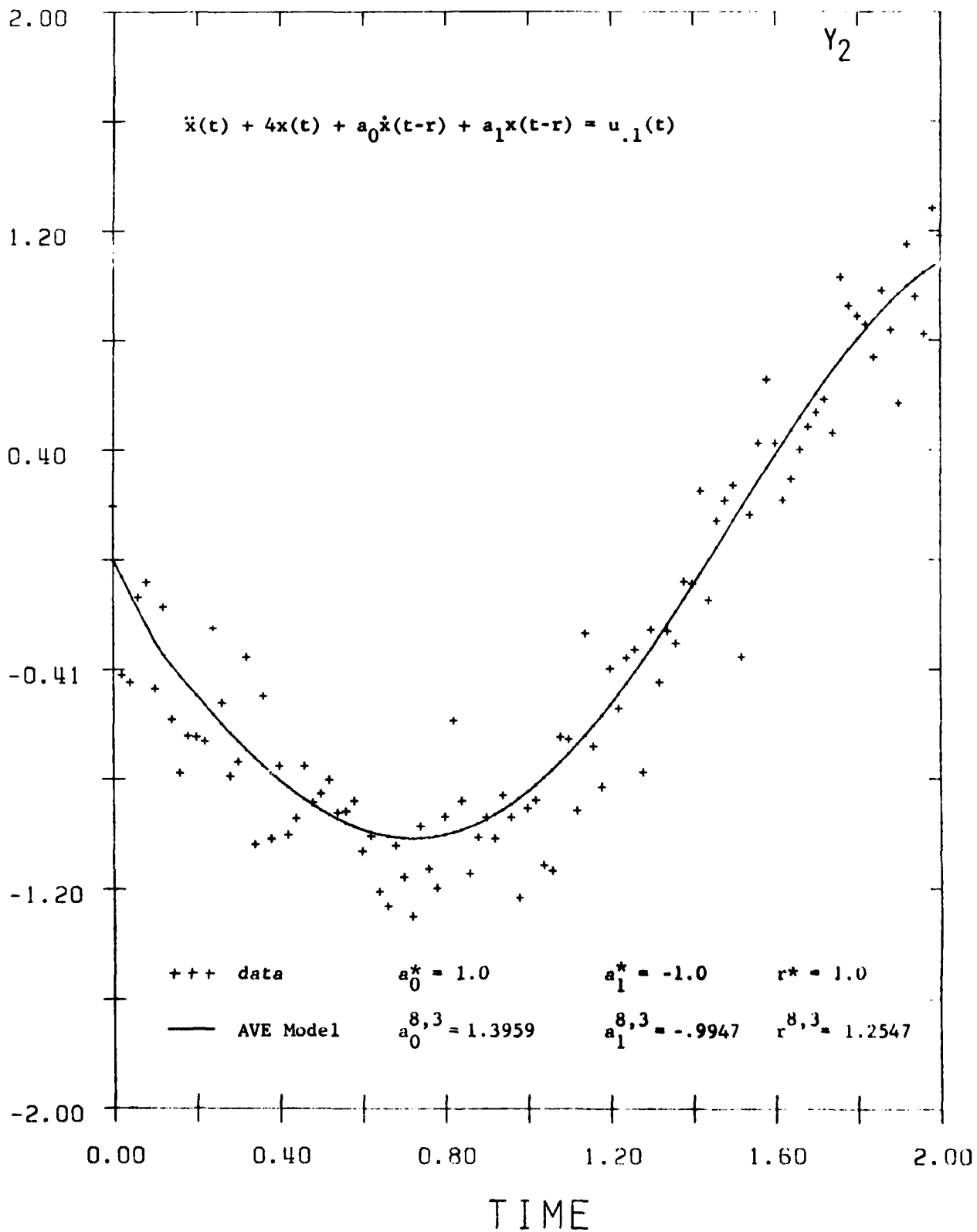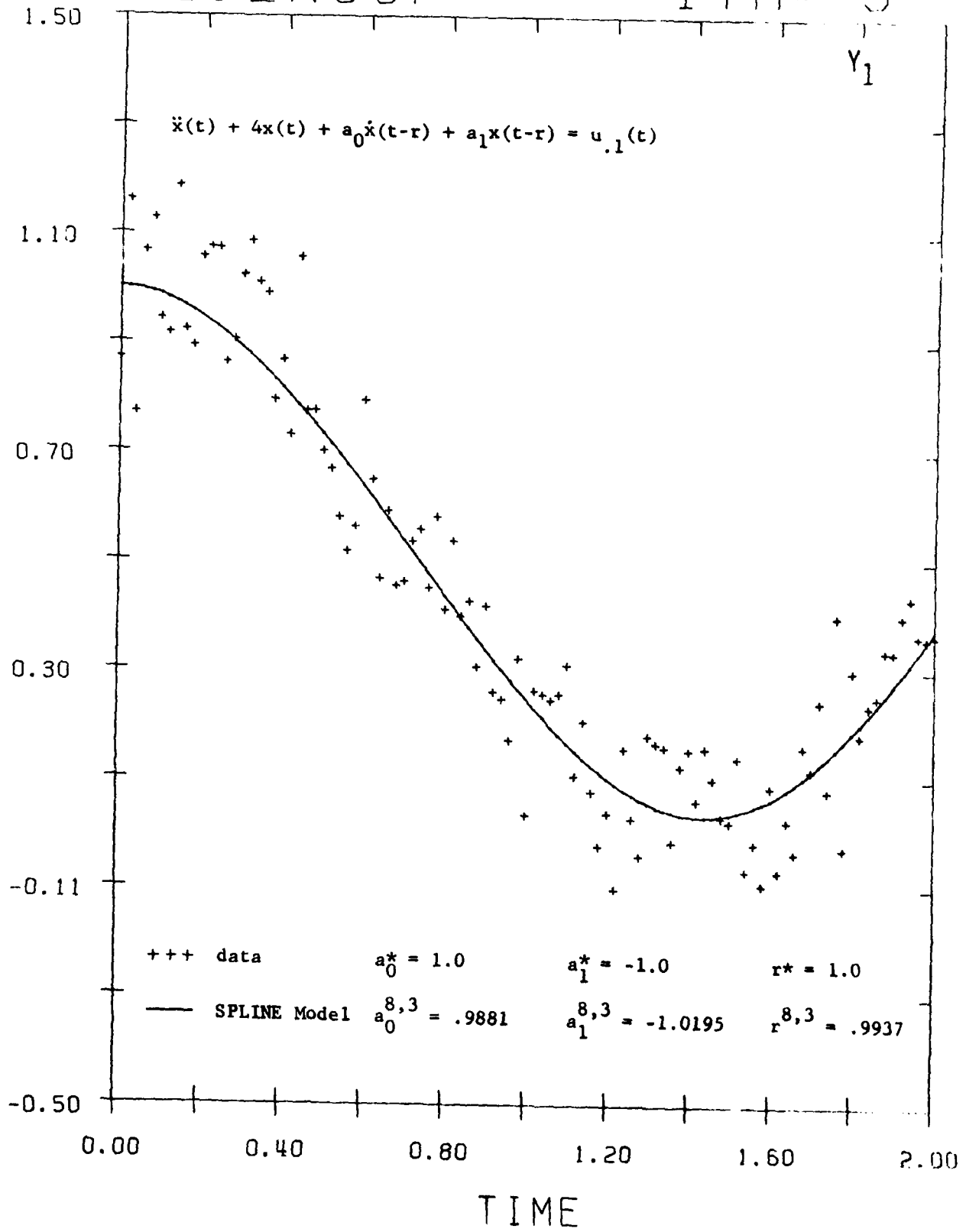
TIME

FIGURE 04.2.2

FIGURE 04.2.3

04.2N8SP   ITR= 3

Y_2

$$\ddot{x}(t) + 4x(t) + a_0\dot{x}(t-r) + a_1x(t-r) = u_{.1}(t)$$

+ + +   data $a_0^* = 1.0$ $a_1^* = -1.0$ $r^* = 1.0$

——— SPLINE Model $a_0^{8,3} = .9881$ $a_1^{8,3} = -1.0195$ $r^{8,3} = .9937$
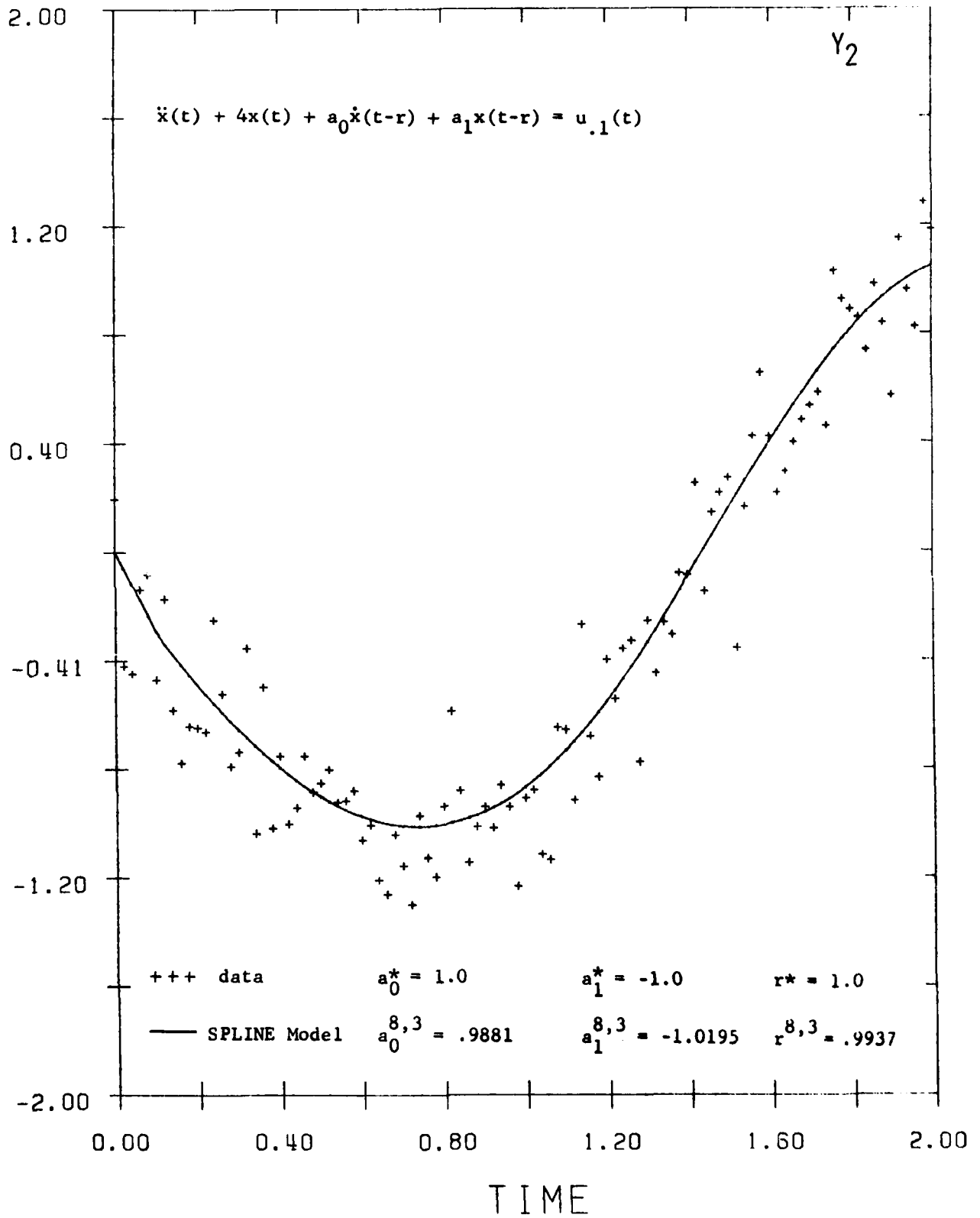
TIME

FIGURE 04.2.4

EXAMPLE 04.3

In this problem all of the system coefficients and the time delay are estimated. In particular the model is assumed to have the form

$$\ddot{x}(t) + \omega^2 x(t) + a_0 \dot{x}(t-r) + a_1 x(t-r) = u_{.1}(t) \ ,$$

with initial data

$$x_0(s) \equiv 1 \ , \ \dot{x}_0(s) \equiv 0 \ , \ -r \le s \le 0 \ ,$$

and vector output

$$y(t) = \begin{bmatrix} x(t) \\ \\ \dot{x}(t) \end{bmatrix}.$$

Start-ups for the true parameter $\omega^* = 2.0$, $a_0^* = 1.0$, $a_1^* = -1.0$, $r^* = 1.0$ were set at

$$\omega^{N,0} = \sqrt{3} \ , \quad a_0^{N,0} = .75 \ , \quad a_1^{N,0} = -.75 \ , \quad r^{N,0} = .8 \ ,$$

and runs were made for $N = 2, 4, 8$ and 16.

For $N = 2$ neither AVE nor SPLINE converged. At $N = 4$ the AVE scheme converged. However, the $N = 4$ MLE procedure for the SPLINE approximation never really converged. The MLE algorithm produced a sequence of parameters that oscillated between two values. These two values are displayed in Table 04.3.2. Observe that either of

the two parameter estimates obtained by the SPLINE scheme is better than the AVE estimate. At N = 8 both AVE and SPLINE converged, while at N = 16 the SPLINE procedure again produced two values for each parameter and the MLE algorithm oscillated between these values.

The data fits for this example are typical of the previous examples. Figures 04.3.1 - 04.3.4 illustrates the N = 8 converged data fits. The data fits at N = 16 for AVE and SPLINE are almost the same, and for the SPLINE scheme either of the two parameters given in Table 04.3.2 produces essentially the same data fits.

AVE					
N	$\hat{\omega}^N$	$\hat{a}_0^N$	$\hat{a}_1^N$	$\hat{r}^N$	$|e_N|$
2	did	not	converge		——
4	1.6475	-2.2109	.1635	.8603	4.7269
8	1.8221	1.6184	-.4164	.8807	1.3799
16	1.7647	1.2959	-.3349	.7405	1.1963
$\gamma^* =$	2.0000	1.0000	-1.0000	1.0000	

TABLE 04.3.1

N	$\hat{\omega}^N$	$\hat{a}_0^N$	$\hat{a}_1^N$	$\hat{r}^N$	$|e_N|$
			SPLINE		
2	did	not	converge		——
4	2.1246	.6354	-1.4428	1.4526	1.3846
4	2.1191	.6633	-1.4263	1.4161	1.2982
8	1.9671	1.0641	- .9164	.9381	.2425
16	1.9436	1.1155	- .8410	.9074	.4235
16	1.9736	1.0529	- .9373	.9573	.1847
$\gamma^* =$	2.0000	1.0000	-1.0000	1.0000	

TABLE 04.3.2

04.3N8AV   ITR= 4

$$\ddot{x}(t) + \omega^2 x(t) + a_0 \dot{x}(t-r) + a_1 x(t-r) = u_{.1}(t)$$

+++ data ω* = 2.0 a_0^* = 1.0 a_1^* = -1.0   r* = 1.0

—— AVE Model $\omega^{8,4}$=1.8221 $a_0^{8,4}$=1.6184 $a_1^{8,4}$=-.4164 $r^{8,4}$=.8807

FIGURE 04.3.1

FIGURE 04.3.2

$$\ddot{x}(t) + \omega^2 x(t) + a_0 \dot{x}(t-r) + a_1 x(t-r) = u_{.1}(t)$$

Y_1

+ + +   data $\omega^* = 2.0$ $a_0^* = 1.0$ $a_1^* = -1.0$ $r^* = 1.0$

———   SPLINE Model $\omega^{8,6} = 1.9671$ $a_0^{8,6} = 1.0641$ $a_1^{8,6} = -.9164$ $r^{8,6} = .9381$

TIME

FIGURE 04.3.3

$$\ddot{x}(t) + \omega^2 x(t) + a_0\dot{x}(t-r) + a_1 x(t-r) = u_{.1}(t)$$

Y_2

+++ data $\omega^* = 2.0$ $a_0^* = 1.0$ $a_1^* = -1.0$ $r^* = 1.0$

——— SPLINE Model $\omega^{8,6} = 1.9671$ $a_0^{8,6} = 1.0641$ $a_1^{8,6} = -.9160$ $r^{8,6} = .9381$
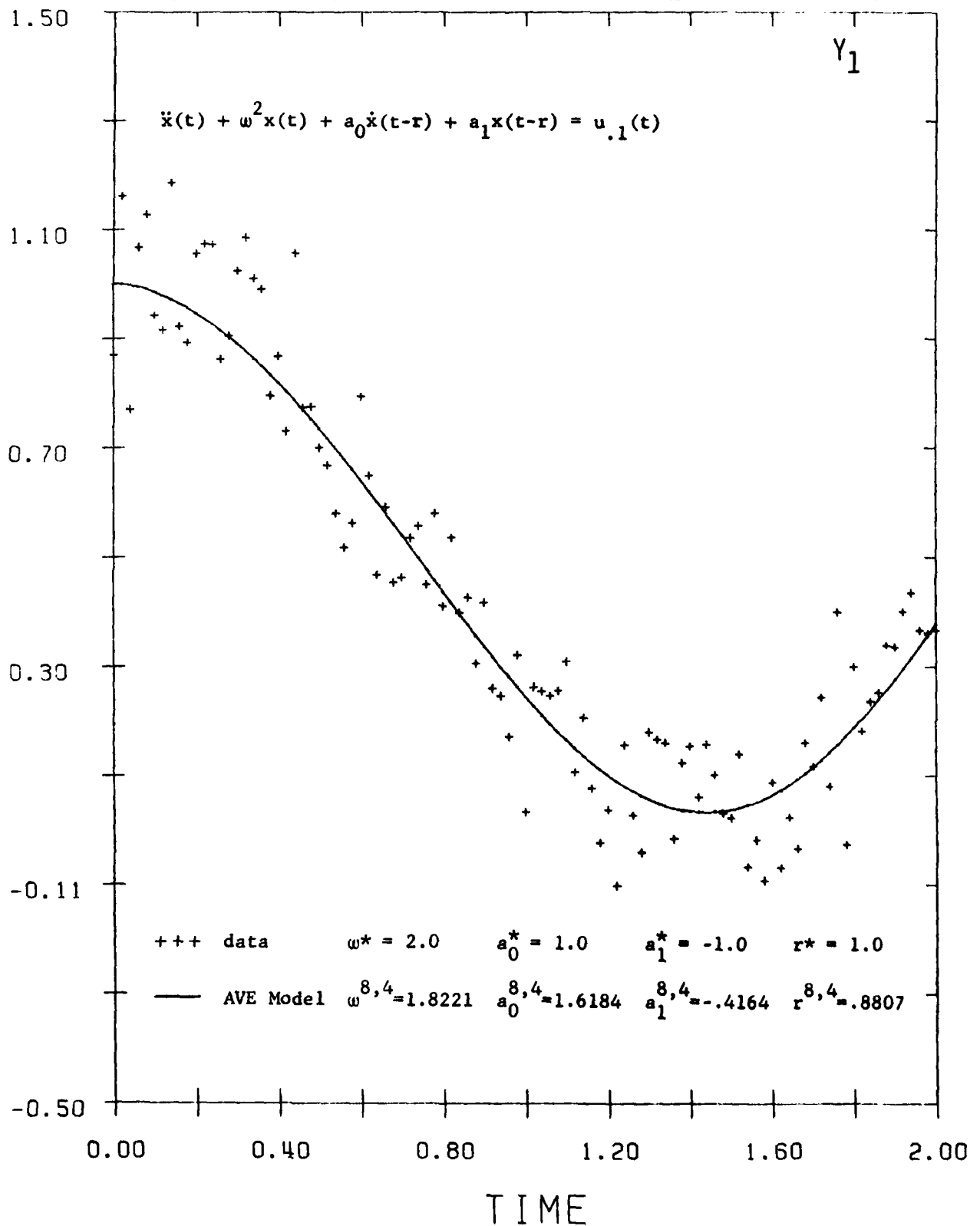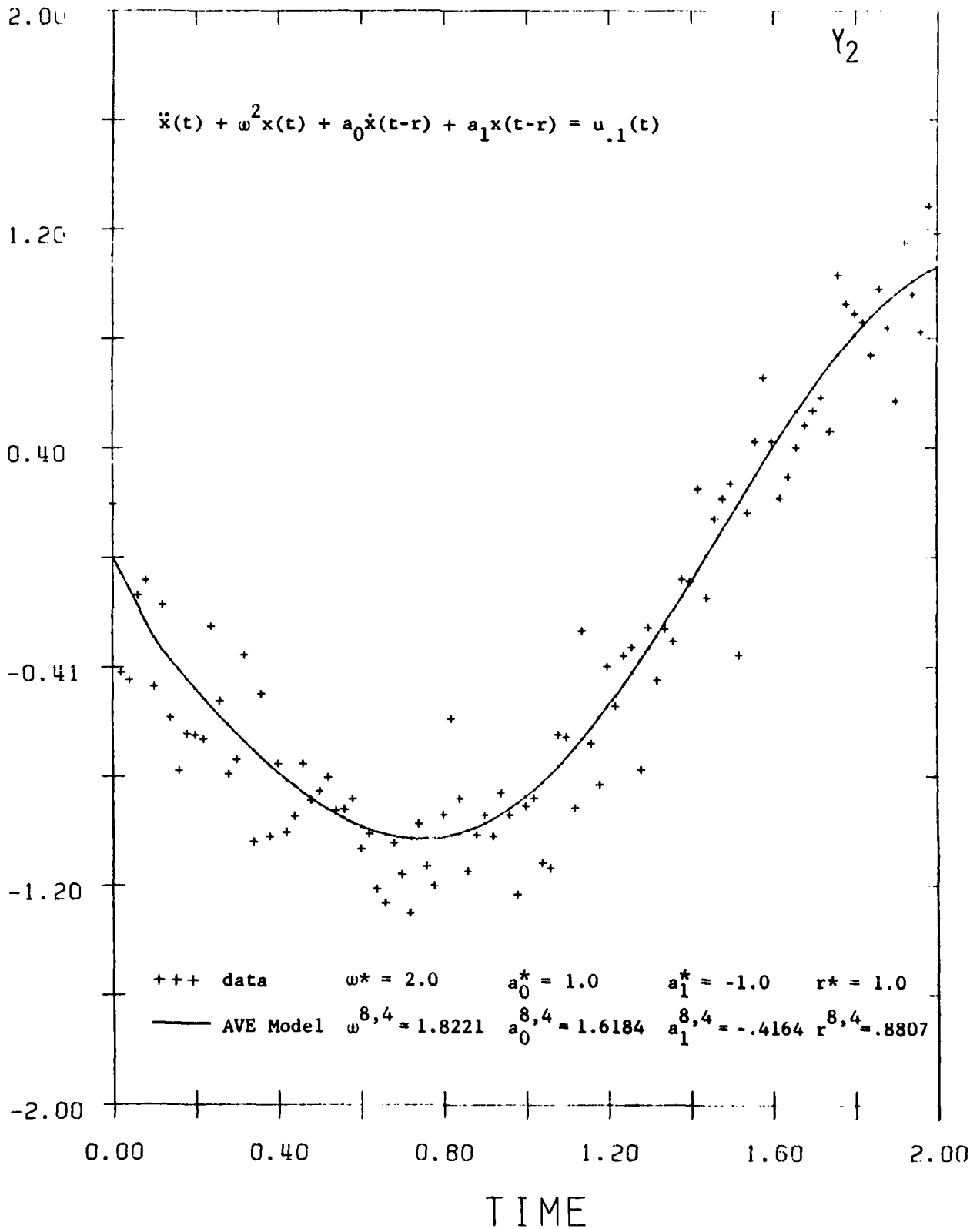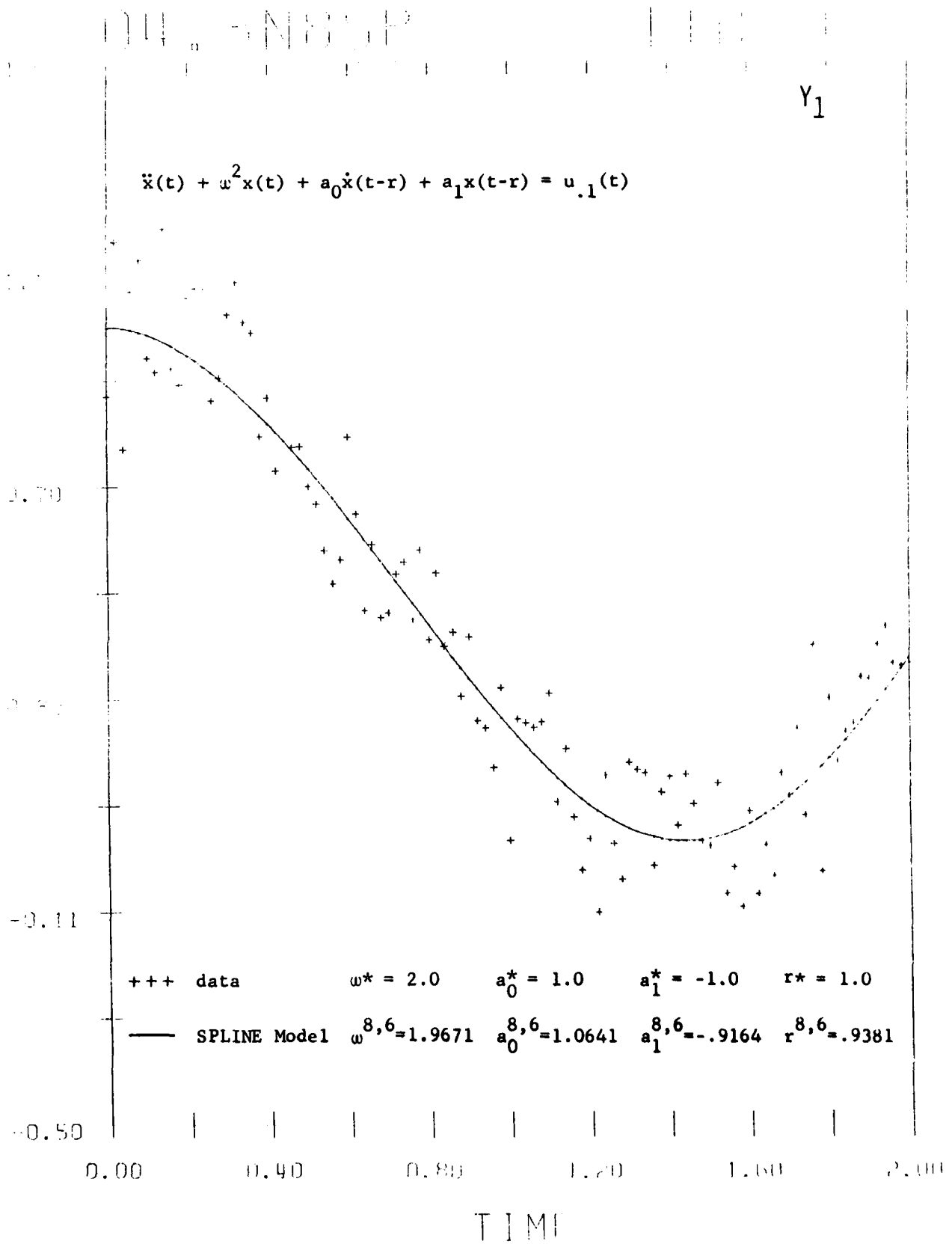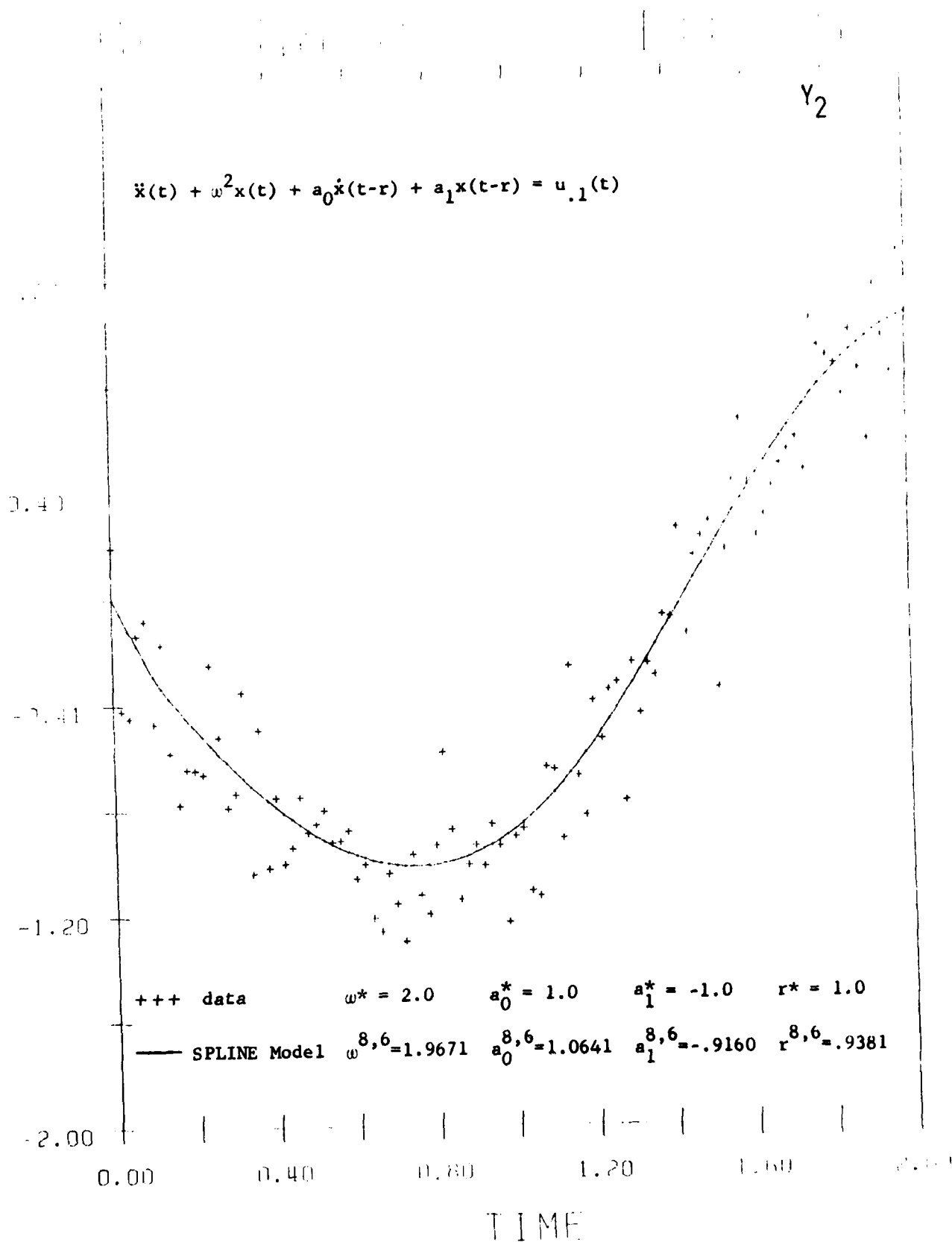
TIME

FIGURE 04.3.4

ID MODEL 05

This model is the second order oscillator considered in ID MODEL 01, with vector output. However, data was generated on a longer time interval by numerically integrating the equations. In particular, the system is governed by the delay equation

$$\ddot{x}(t) + 36x(t) + 2.5\dot{x}(t-1) + 9x(t-1) = u_{.1}(t) ,$$

with initial data

$$x_0(s) \equiv 1 , \quad \dot{x}_0(s) \equiv 0 , \quad -1 \leq s \leq 0 ,$$

and vector output

$$y(t) = \begin{bmatrix} y_1(t) \\ \\ y_2(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ \\ \dot{x}(t) \end{bmatrix} .$$

This system was numerically integrated (using a modified 4th order scheme) to obtain the solution on the interval $[0,5]$. Data was generated at 101 equally spaced points (i.e. 20 data points per unit interval) using this numerical solution.

As a rough check of the numerically produced data, the numerical solution and the analytic solution were compared on the interval $[0,2]$. The numerical solution agreed exactly (i.e. to eight decimal places) with the analytic solution, giving some indication that the data for this model is reasonably good.

As a final comment, we mention that the numerical algorithm used to integrate the delay equation is completely unrelated to any of the approximation schemes used in the identification algorithms. Consequently, we are not using data generated by the algorithm that we are attempting to study.

EXAMPLE 05.1

For this example we seek to estimate the time delay and two system coefficients. The model is assumed to be of the form

$$\ddot{x}(t) + 36x(t) + a_0\dot{x}(t-r) + a_1 x(t-r) = u_{.1}(t) \ ,$$

with initial data

$$x_0(s) = 1 \ , \quad \dot{x}_0(s) \equiv 0 \ , \quad -r \leq s \leq 0 \ ,$$

and vector output

$$y(t) = \begin{bmatrix} x(t) \\ \\ \dot{x}(t) \end{bmatrix} \ .$$

Recall that in this example we have data generated for 5 seconds, i.e. on the interval $[0,5]$, at 101 points. The parameters to be estimated are $r^* = 1.0$, $a_0^* = 2.5$ and $a_1^* = 9.0$. Start-up values for each run were

$$r^{N,0} = .9 \ , \quad a_0^{N,0} = 2.2 \ , \quad a_1^{N,0} = 9.5 \ .$$

Other start-up values were attempted and the algorithms were found to diverge if the start-up errors were too large and in some cases the algorithms converged to parameters different than r^*, a_0^*, a_1^*. This again shows that there can be a lack of global identifiability.

Runs for $N = 2$, 4, 8 and 16 were made for both AVE and SPLINE.

For low N, the AVE scheme did not converge. However, both AVE and SPLINE converged for larger N and produced reasonable estimates of the parameters. These results are summarized in Tables 05.1.1 and 05.1.2. Observe that at N = 16 the SPLINE procedure produced estimates with approximately 2% relative ℓ_1 error, while the N = 16 AVE estimates have relative ℓ_1 error greater than 22%.

The data fits for this example are very interesting. This example is very dynamic and oscillatory on the interval [0,5]. However, at N = 16 both AVE and SPLINE produce fairly good data fits, with the SPLINE scheme matching the data better than AVE. Figures 05.1.1 - 05.1.4 show the iteration 0 and converged data fits for the AVE scheme. Figures 05.1.5 - 05.1.8 illustrate the same thing for the SPLINE procedure.

	AVE			
N	$\hat{r}^N$	$\hat{a}_0^N$	$\hat{a}_1^N$	$|e_N|$
2	did not converge			—
4	did not converge			—
8	.2492	2.8002	- 3.8982	13.9492
16	.9106	1.7439	10.9570	2.8025
$\gamma^* =$	1.0000	2.5000	9.0000	

TABLE 05.1.1

		SPLINE		
N	$\hat{r}^N$	$\hat{a}_0^N$	$\hat{a}_1^N$	$|e_N|$
2	did not converge			——
4	.6812	-2.3261	11.7017	7.8466
8	.9985	2.9163	8.9459	.4719
16	1.0000	2.6016	9.0872	.1888
$\gamma^* =$	1.0000	2.5000	9.0000	

TABLE 05.1.2
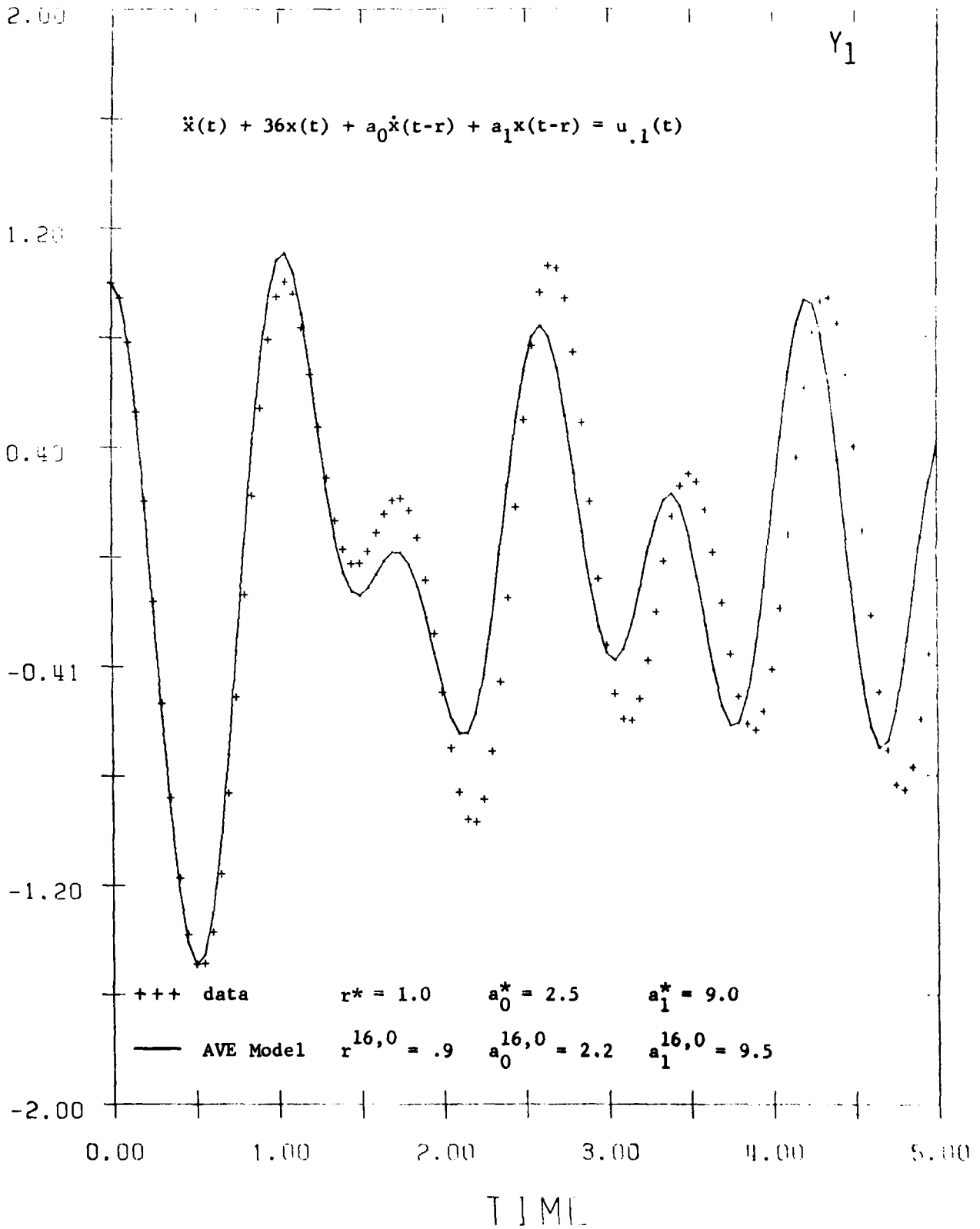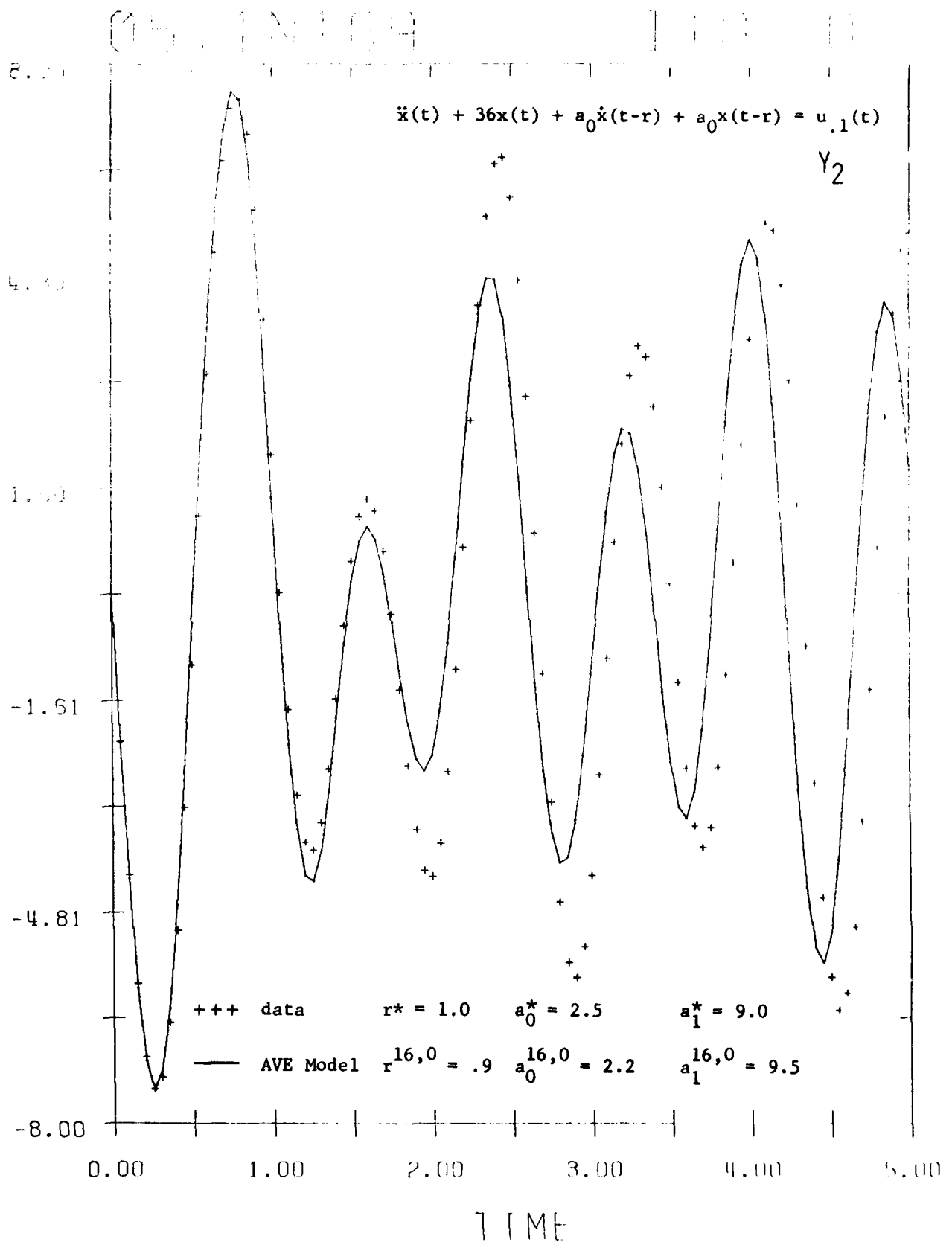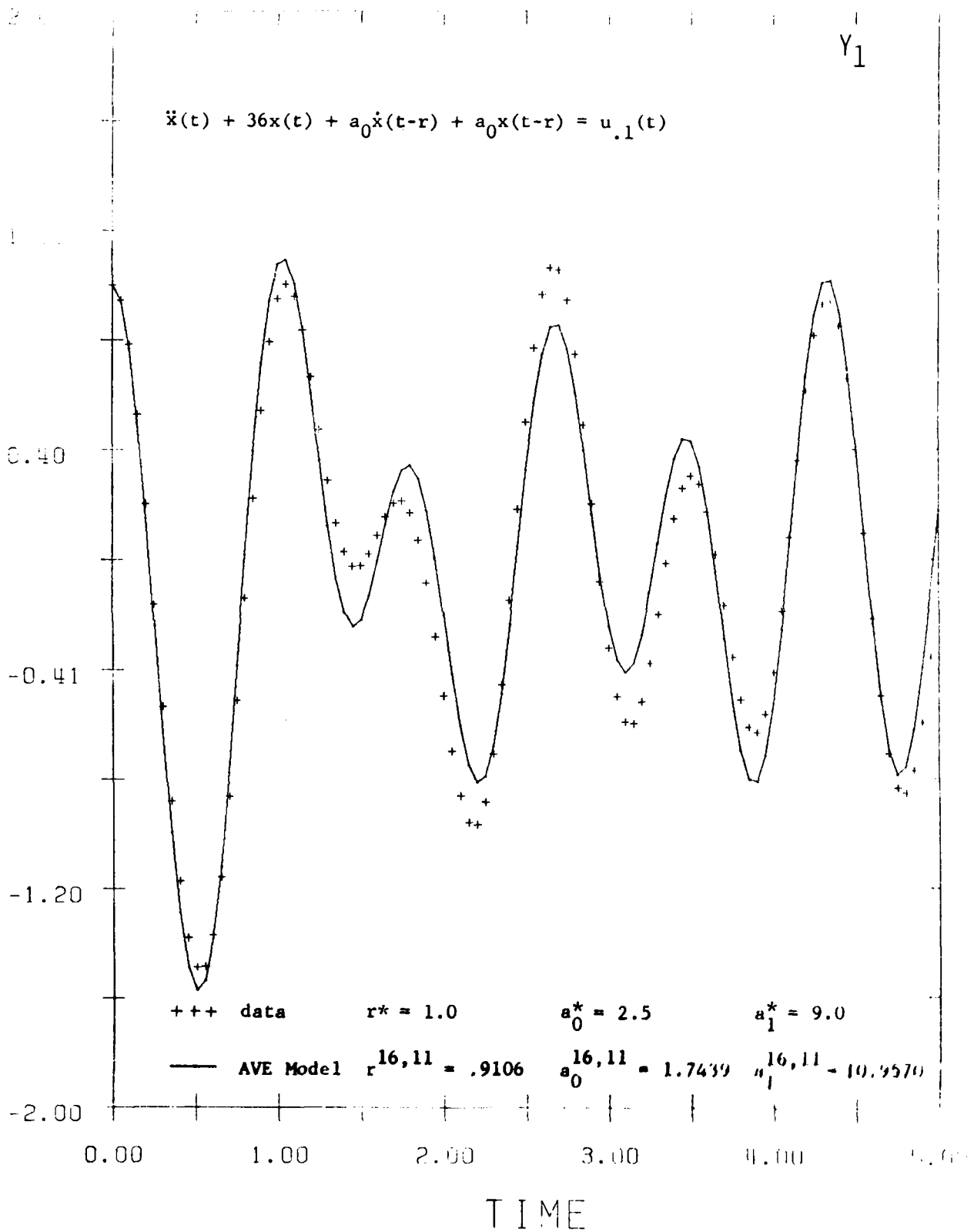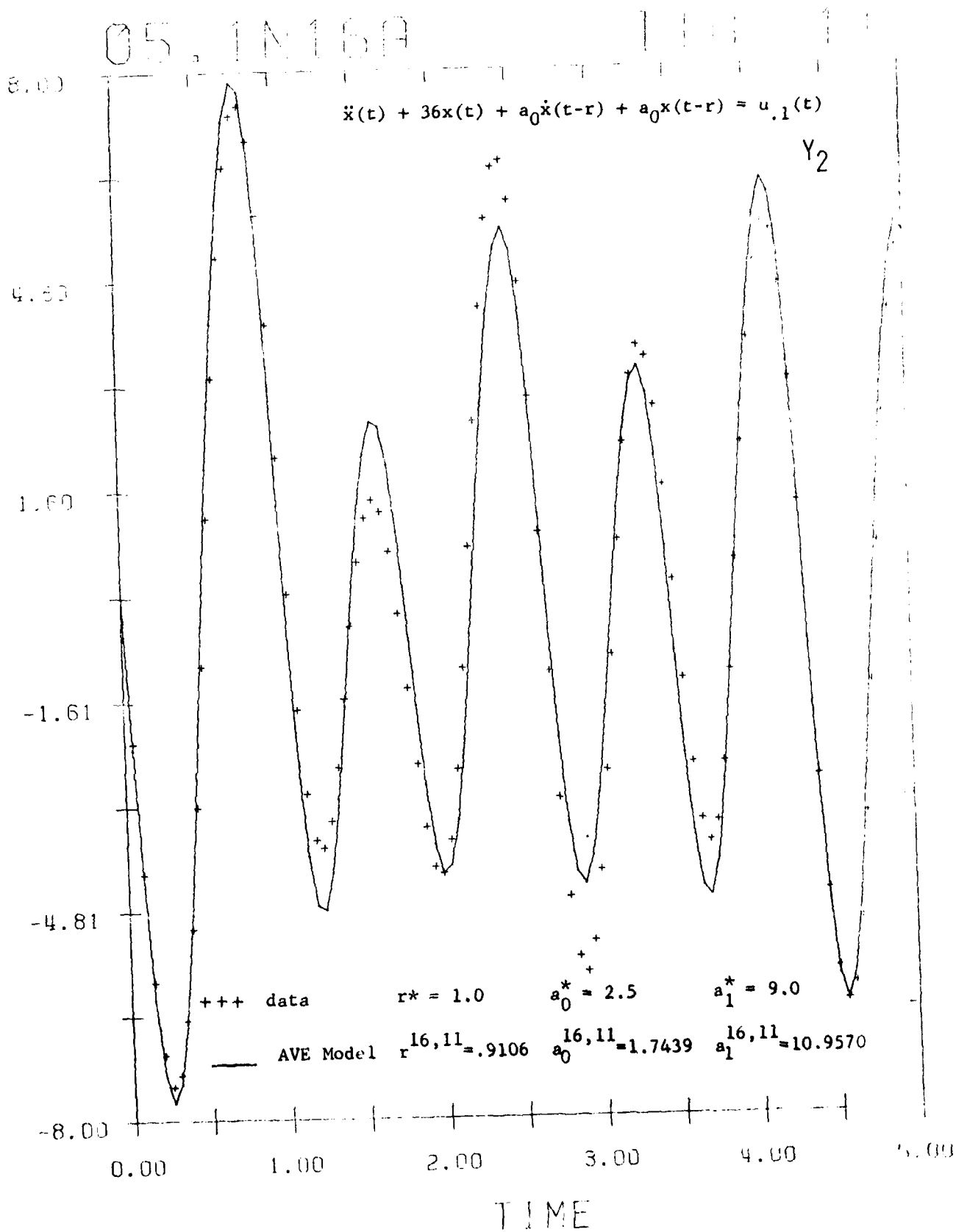
FIGURE 05.1.1

$$\ddot{x}(t) + 36x(t) + a_0\dot{x}(t-r) + a_0 x(t-r) = u_{.1}(t)$$

Y_2

+ + + data $r^* = 1.0$ $a_0^* = 2.5$ $a_1^* = 9.0$

——— AVE Model $r^{16,0} = .9$ $a_0^{16,0} = 2.2$ $a_1^{16,0} = 9.5$

TIME

FIGURE 05.1.2

$$\ddot{x}(t) + 36x(t) + a_0\dot{x}(t-r) + a_0x(t-r) = u_{.1}(t)$$

FIGURE 05.1.3

$$\ddot{x}(t) + 36x(t) + a_0\dot{x}(t-r) + a_0x(t-r) = u_{.1}(t)$$

Y_2

+ + + data $r* = 1.0$ $a_0^* = 2.5$ $a_1^* = 9.0$

——— AVE Model $r^{16,11} = .9106$ $a_0^{16,11} = 1.7439$ $a_1^{16,11} = 10.9570$

TIME

FIGURE 05.1.4

$$\ddot{x}(t) + 36x(t) + a_0\dot{x}(t-r) + a_0x(t-r) = u_{.1}(t)$$

+++   data $r^* = 1.0$ $a_0^* = 2.5$ $a_1^* = 9.0$

——   SPLINE Model $r^{16,0} = .9$ $a_0^{16,0} = 2.2$ $a_1^{16,0} = 9.5$

TIME

FIGURE 05.1.5

$$\ddot{x}(t) + 36x(t) + a_0\dot{x}(t-r) + a_0x(t-r) = u_{.1}(t)$$

Y_2

+ + +   data $r* = 1.0$ $a_0^* = 2.5$ $a_1^* = 9.0$

——— SPLINE Model $r^{16,0} = .9$ $a_0^{16,0} = 2.2$ $a_1^{16,0} = 9.5$

TIME

FIGURE 05.1.6

$$\ddot{x}(t) + 36x(t) + a_0\dot{x}(t-r) + a_0 x(t-r) = u_{.1}(t)$$

+++ data	$r* = 1.0$	$a_0^* = 2.5$	$a_1^* = 9.0$
—— SPLINE Model	$r^{16,8} = 1.0000$	$a_0^{16,8} = 2.6016$	$a_1^{16,8} = 9.0872$
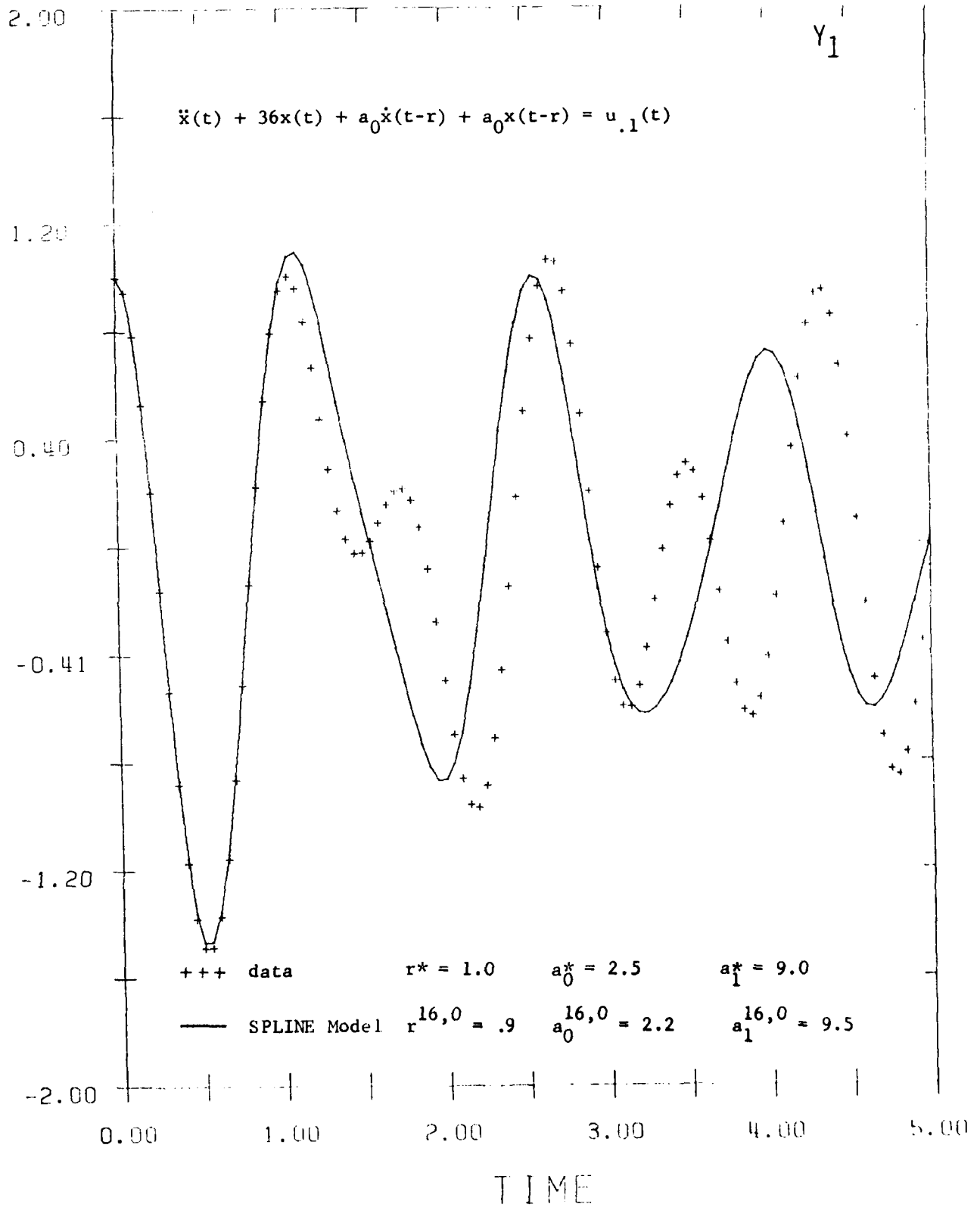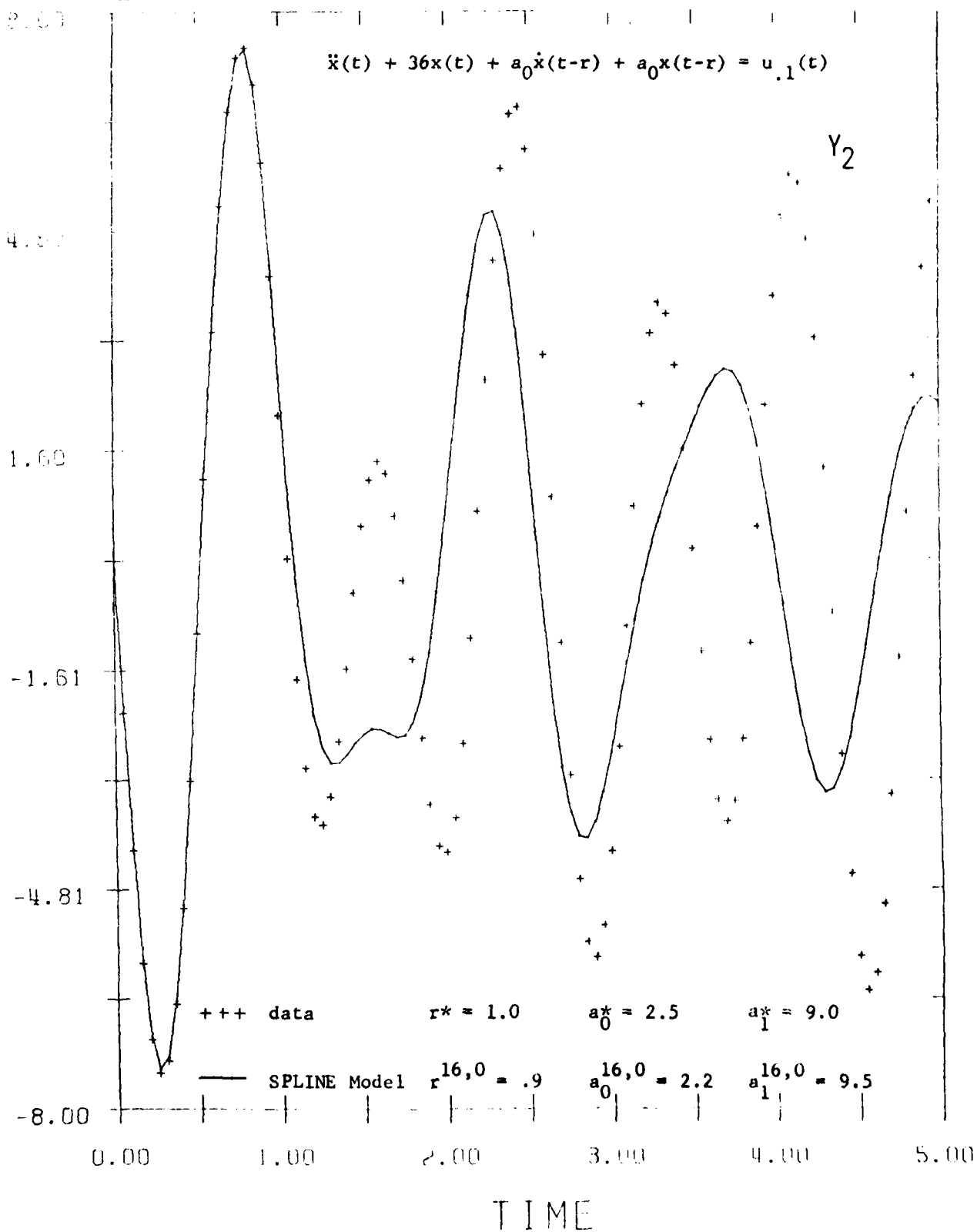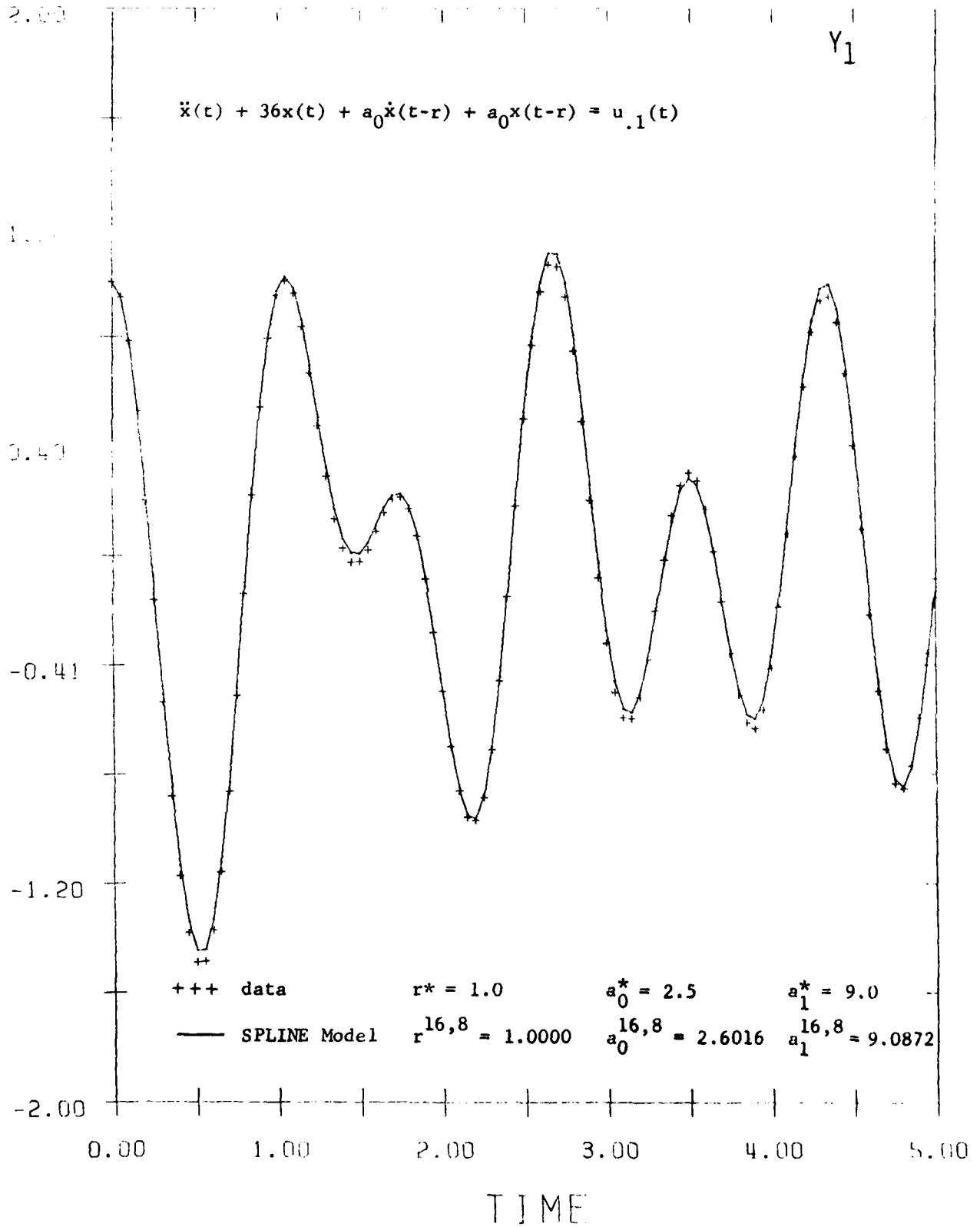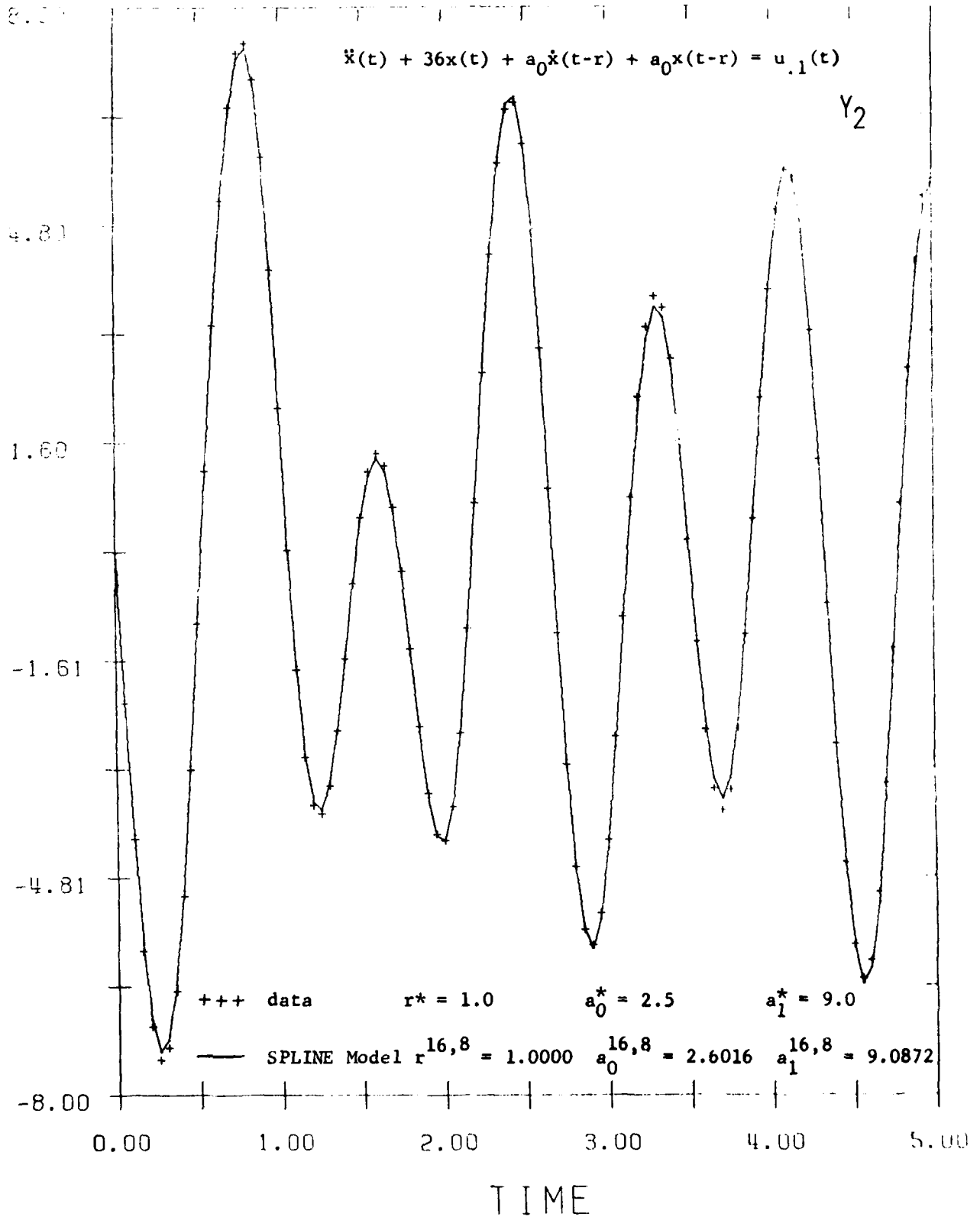
FIGURE 05.1.7

FIGURE 05.1.8

ID MODEL K1

This is a model with continuous kernel. The system is governed by the equation

$$\ddot{x}(t) = -3x(t-1) - \int_{-1}^{0} x(t+s)ds + u_{.1}(t)$$

with initial data

$$x_0(s) \equiv 1 , \quad -1 \leq s \leq 0 ,$$

and output

$$y(t) = x(t) .$$

This equation can be transformed to a system of two equations with no integral term (see pages 63-64 of [9]) and solved using the method of steps. The analytic solution was obtained by this procedure and data was generated at 101 equally spaced points on [0,2]. This data was used in the following examples; K1.1, K1.4.

EXAMPLE K1.1

In this example we attempt to identify the kernel by assuming
that it is constant function with unknown value. Therefore, the
model is of the form

$$\dot{x}(t) = -3x(t-1) + k \int_{-1}^{0} x(t+s)\,ds + u_{.1}(t) \ ,$$

with initial data

$$x_0(t) \equiv 1 \ , \quad -1 \leq s \leq 0 \ ,$$

and output

$$y(t) = x(t) \ .$$

The constant $k* = -1.0$ is to be estimated. Runs at $N = 2,4,8,16$
were made with the start-up of

$$k^{N,0} = .0 \ .$$

This example is interesting for several reasons. It is an
example that contains a distributed delay and it is the only
example we have run where the AVE scheme produced better parameter
estimates than the SPLINE scheme. Table K1.1.1 illustrates the
convergence of the parameter estimates for AVE and SPLINE. Figures
K1.1.1 and K1.1.2 compare the $N = 8$ converged data fits for AVE and
SPLINE, respectively. Observe that even though the $N = 8$ AVE scheme
produced a better parameter estimate, the $N = 8$ SPLINE scheme does a

much better job of fitting the data.

AVE			SPLINE		
N	$\hat{k}^N$	$\|e_N\|$	N	$\hat{k}^N$	$\|e_N\|$
2	-1.2953	.2953	2	-1.2679	.2679
4	-1.0765	.0765	4	-1.0827	.0827
8	-1.0156	.0156	8	-1.0301	.0301
16	-1.0058	.0058	16	-1.0177	.0177
$k^* = -1.0000$			$k^* = -1.0000$		

TABLE K1.1.1

$$\dot{x}(t) = -3x(t-1) + k \int_{-1}^{0} x(t+s)\ ds + u_{.1}(t)$$

+ + + data $\qquad k* = -1.0$

—— AVE Model $k^{8,6} = -1.0156$
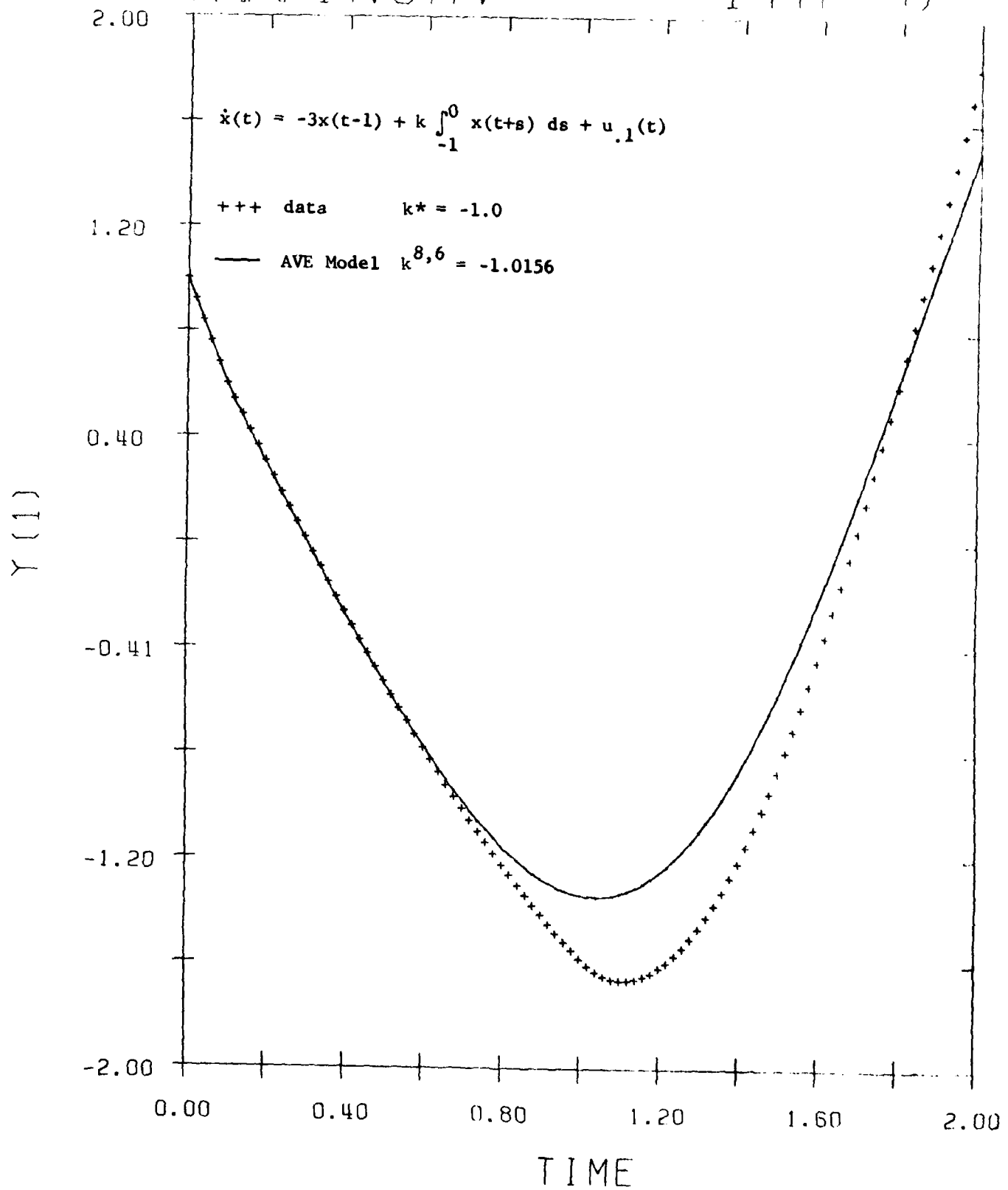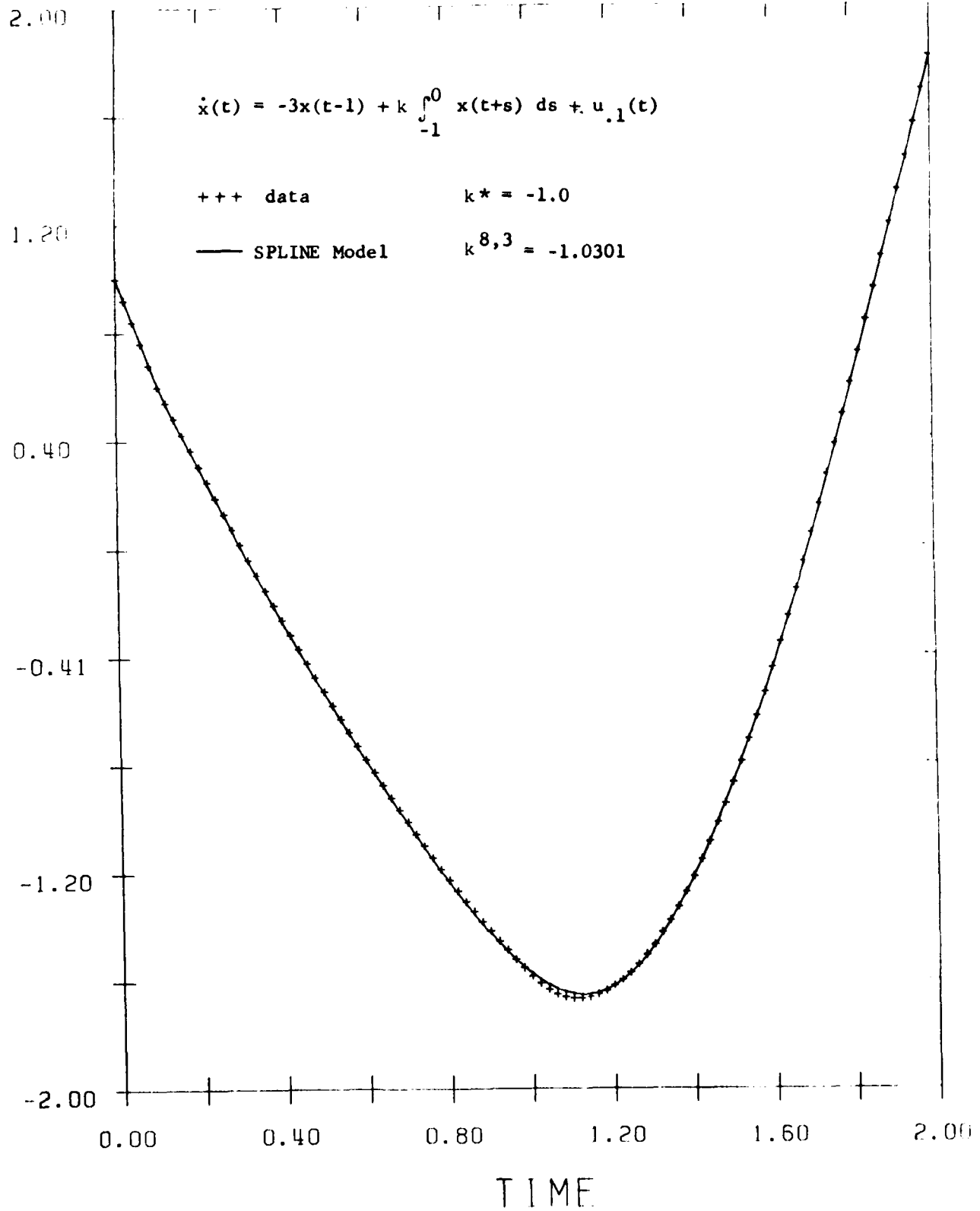
FIGURE K1.1.1

FIGURE K1.1.2

EXAMPLE K1.4

In this example we estimate two system coefficients and the time delay. As before, the kernel is estimated by assuming that it is an unknown but constant function. Therefore, the model is assumed to be of the form

$$\dot{x}(t) = a_1 x(t-r) + k \int_{-r}^{0} x(t+s) \, ds + u_{.1}(t) \; ,$$

with initial data

$$x_0(s) \equiv 1 \; , \quad -r \le s \le 0 \; ,$$

and output

$$y(t) = x(t) \; .$$

The true parameters $a_1^* = -3.0$, $k^* = -1.0$ and $r^* = 1.0$ we estimated using start-ups of

$$a_1^{N,0} = -3.5 \; , \quad k^{N,0} = -1.5 \; , \quad r^{N,0} = 1.5 \; .$$

Runs were made for N = 2,4,8 and 16. The AVE scheme did not converge for N = 2 and 4. However, for N = 8 and 16 the AVE scheme converged but produced rather poor parameter estimates. The SPLINE scheme converged for each N = 2,4,8,16 and for N ≥ 4 produced good parameter estimates. The numerical results for this problem are summarized in Tables K1.4.1 and K1.4.2.

Figures K1.4.1 - K1.4.4 compare the N = 8 AVE and SPLINE data

fits.   In particular, Figures K1.4.1 and K1.4.2 show the N = 8 AVE
start-up and converged data fits, respectively.   Figures K1.4.3
and K1.4.4 show the same thing for the SPLINE procedure.

AVE				
N	$\hat{r}^N$	$\hat{k}^N$	$\hat{a}_1^N$	$|e_N|$
2	did	not	converge	——
4	did	not	converge	——
8	.8802	.2182	-4.1641	2.0657
16	.9383	-.3806	-3.5535	1.2346
γ^* =	1.0000	-1.0000	-3.0000	

TABLE K1.4.1

SPLINE				
N	$\hat{r}^N$	$\hat{k}^N$	$\hat{a}_1^N$	$|eN|$
2	.9100	- .4376	-3.4478	1.1002
4	.9896	-1.0087	-3.0580	.0771
8	1.0018	-1.0390	-2.9953	.0455
16	1.0042	-1.0410	-2.9841	.0611
γ^* =	1.0000	-1.0001	-3.0000	

TABLE K1.4.1

K1.4N8AV   ITR 0
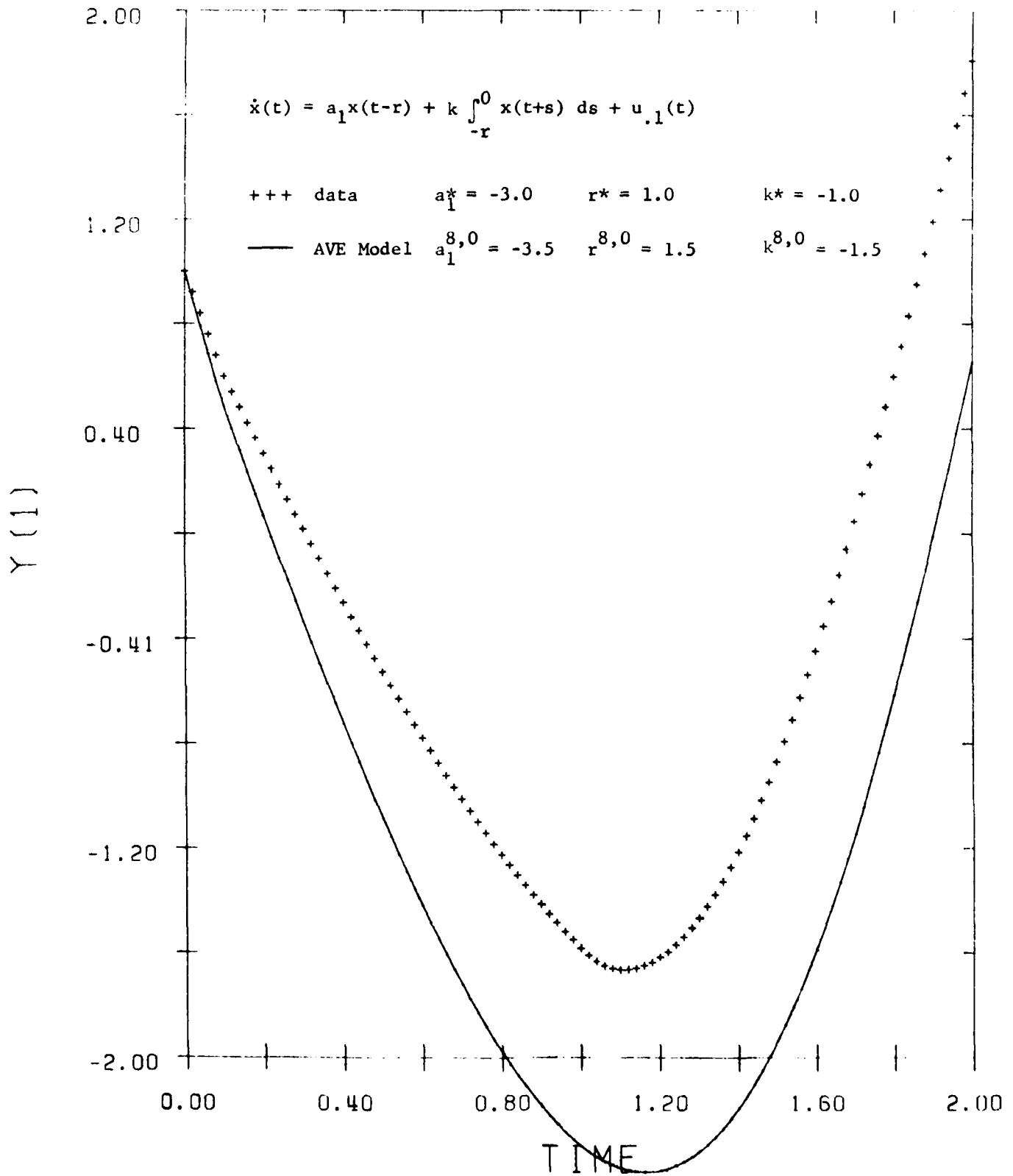
$$\dot{x}(t) = a_1 x(t-r) + k \int_{-r}^{0} x(t+s) \, ds + u_{.1}(t)$$

+ + + data $a_1^* = -3.0$ $r^* = 1.0$ $k^* = -1.0$

——— AVE Model $a_1^{8,0} = -3.5$ $r^{8,0} = 1.5$ $k^{8,0} = -1.5$

FIGURE K1.4.1

FIGURE K1.4.2

K1.4N8SP   TIR 0

$$\dot{x}(t) = a_1 x(t-r) + k \int_{-r}^{0} x(t+s) \, ds + u_{.1}(t)$$

+ + + data $a_1^* = -3.0$ $r^* = 1.0$ $k^* = -1.0$

——— SPLINE Model $a_1^{8,0} = -3.5$ $r^{8,0} = 1.5$ $k^{8,0} = -1.5$

Y(1)

TIME
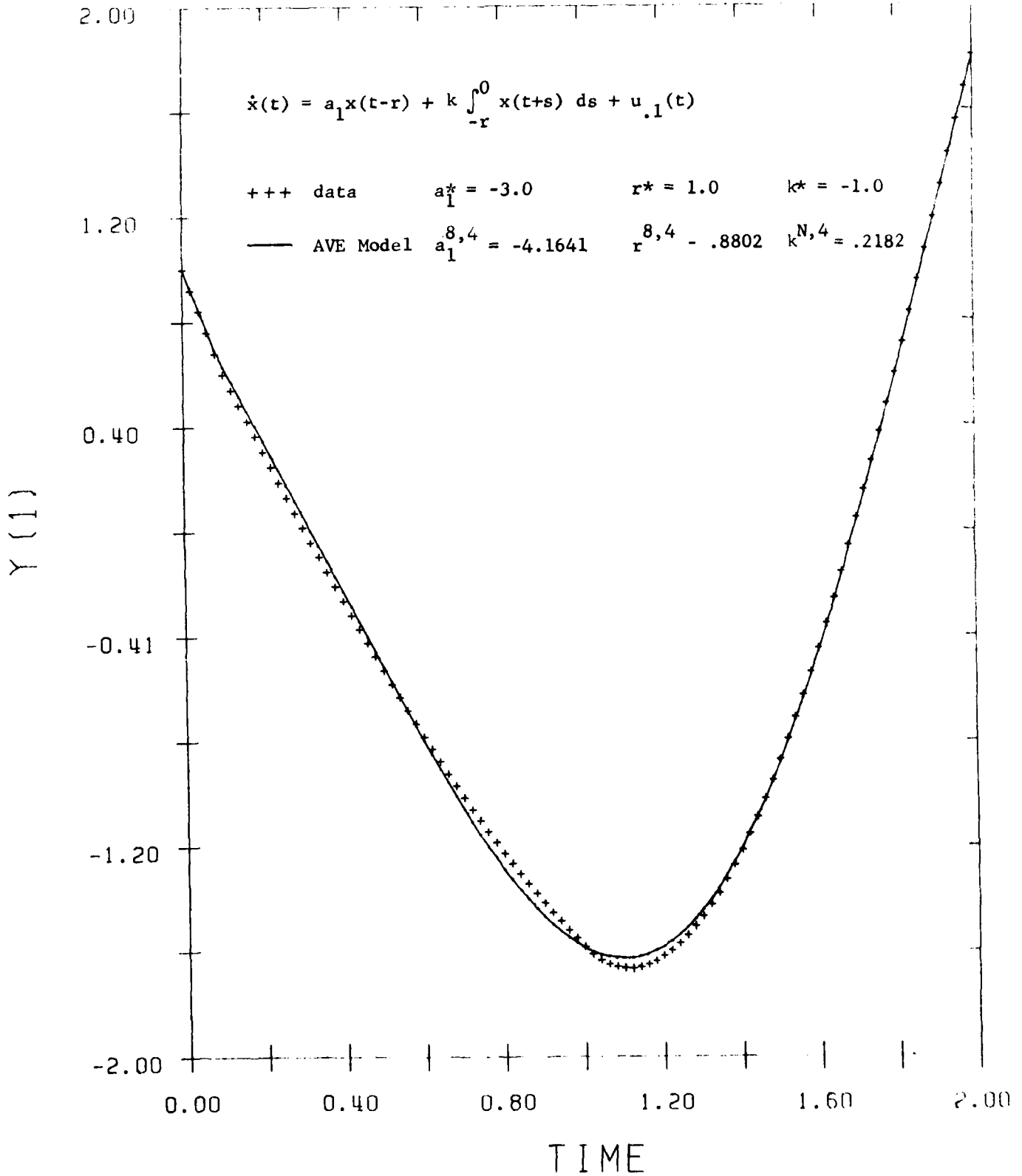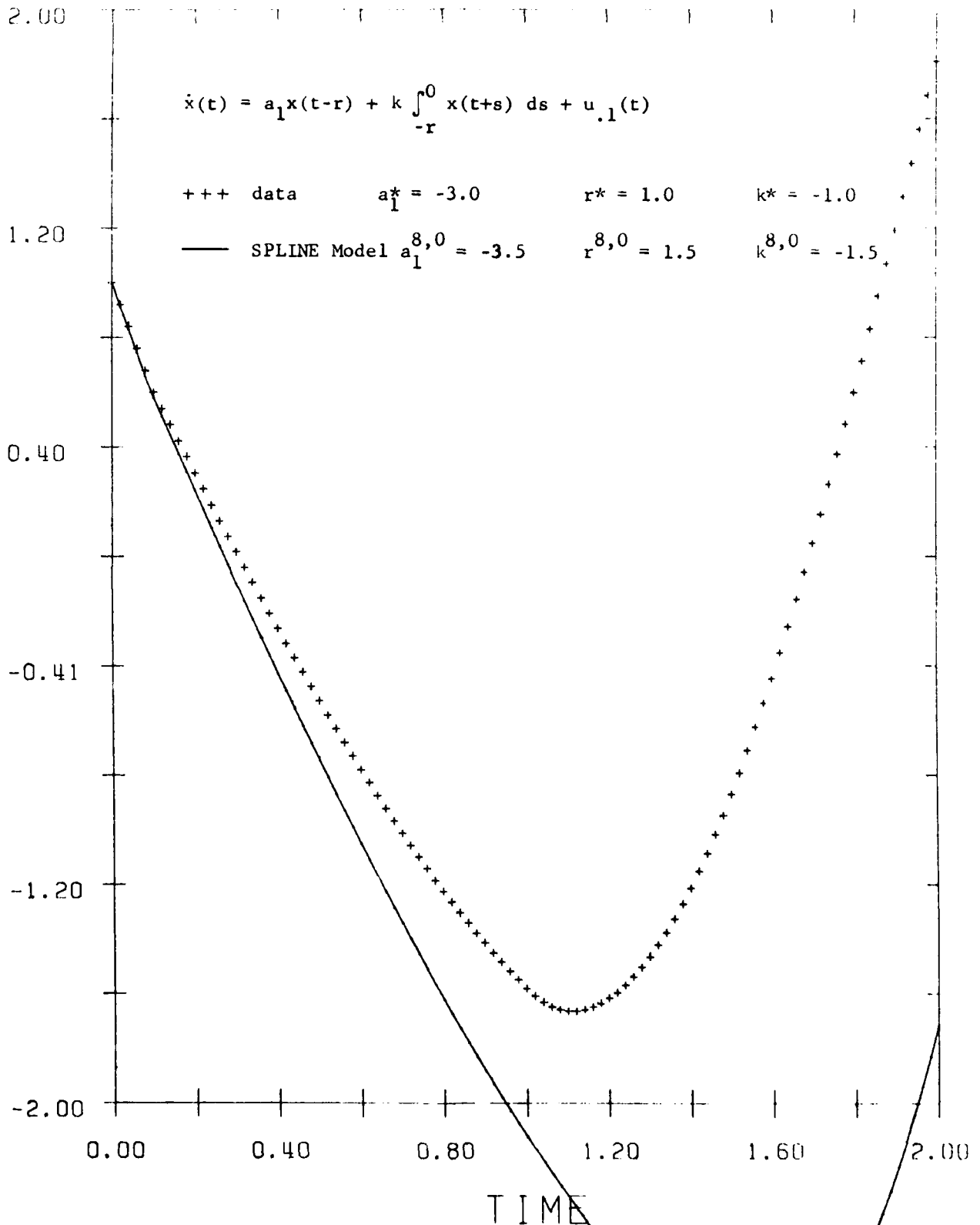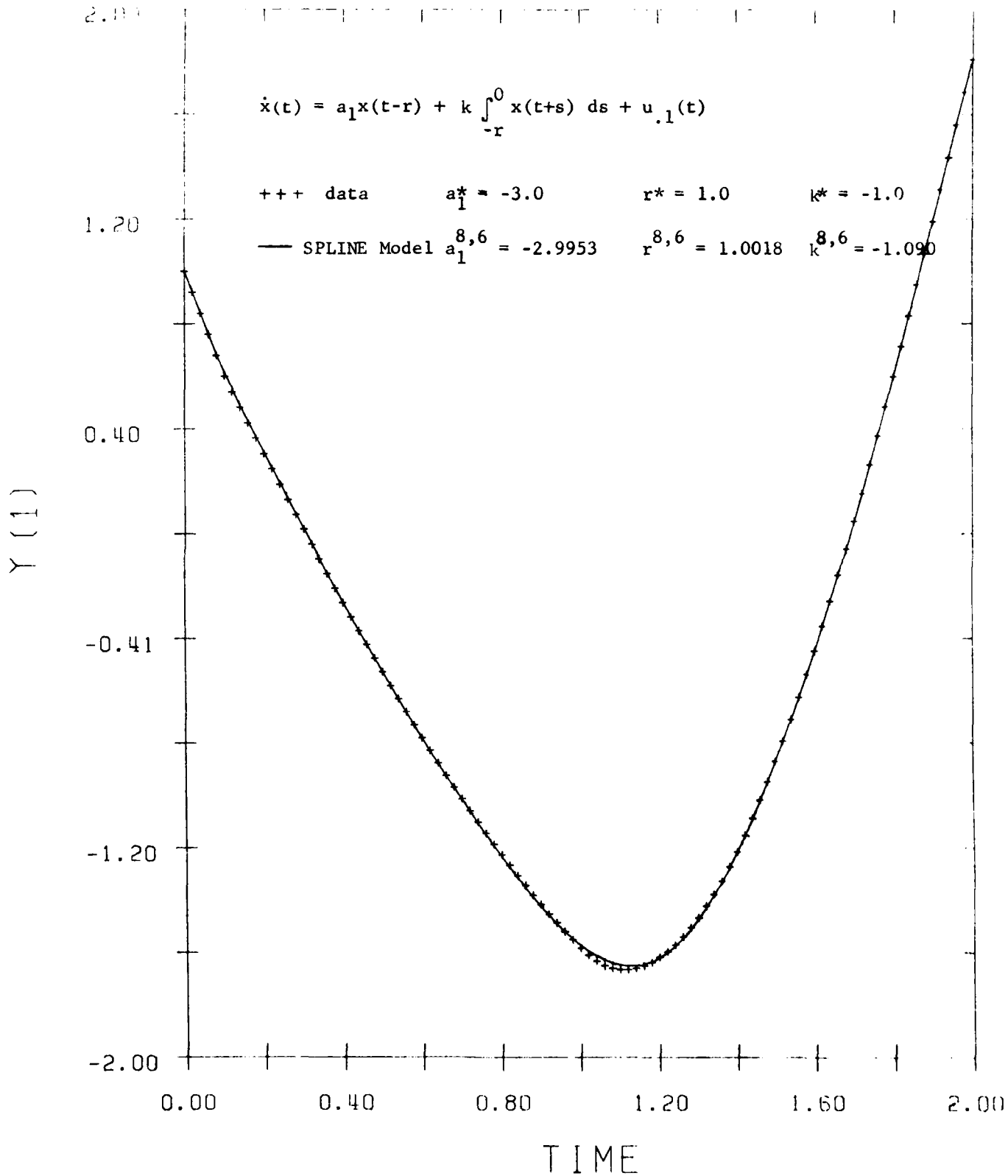
FIGURE K1.4.3

FIGURE K1.4.4

Summary Remarks

The major lesson of our experience with the AVE and SPLINE
identification procedures is that SPLINE is generally superior and
commonly displays near quadratic convergence. It has been observed
that the error function E^N can have multiple relative minima. In
order to solve the IDN problem we have used a maximum likelihood
estimator (MLE), which in the case of a scalar measurement, is
equivalent to the usual quasi-linerization (QL) procedure for
minimizing E^N. Conditions that guarantee convergence of the QL
procedure are rather stringent (see [7]) and, in fact, are not
satisfied in our examples. In applications it would seem prudent
to employ a hybrid algorithm for the IDN problem, wherein one
would initially use a method that guarantees descent and then
employ QL only in the neighborhood of a minimizing point.

6. <u>The optimal control problem: numerical examples</u>.

In this section we present numerical results for a number of
examples of the optimal control problem (OC). The two schemes
AVE and SPLINE were employed to compute approximating optimal
controls $\hat{u}_A^N$ and $\hat{u}_S^N$ for several values of N chosen to illustrate
convergence properties. The linear systems examples given here
are essentially the same as some of the examples considered in
detail in [6]. For these examples the analytic solution u* of the
optimal control problems can be found in §3 of that report and we
shall not redrive those solutions here. Only one (C8) of the
linear examples presented below was not considered specifically
in [6]; however for the particular case detailed here the optimal
control can be computed by using the maximum principle for delay
systems in the same manner as was done for Example 10 of [6].
Since the report [6] is rather complete and easily obtained, we
shall feel free to use the results presented in that technical
report without elaborate comment or discussion. For the motivation
behind our choice of some of the particular examples presented
here and in [6], the interested reader can consult [6].

We also present below our numerical findings for two nonlinear
examples. The theory for use of the AVE approximations with a
restricted but reasonable class of nonlinear system optimal control
problems is developed in [1]. Consideration of the arguments given
there along with details of the SPLINE scheme development in [8]

should convince the reader that a corresponding theory for the

SPLINE scheme can be developed in a straightforward manner.

EXAMPLE C1

The first system is given by the scalar equation

$$\dot{x}(t) = x(t - 1) + u(t), \quad 0 \le t \le 3$$

with initial data

$$x(s) \equiv 1, \quad -1 \le s \le 0 ,$$

and the payoff is chosen as

$$J(u) = 5[x(3)]^2 + \tfrac{1}{2} \int_0^3 [u(s)]^2 \, ds .$$

The optimal control u* is given by (see page 11 in [6]

$$u^*(t) = \begin{cases} \delta\{-(t - 2)^2/2 - 3/2\}, & 0 \le t \le 1, \\ \delta(t - 3) & , 1 \le t \le 2 \\ -\delta & , 2 \le t \le 3, \end{cases}$$

where $\delta = 370/\{6(1 + 319/3)\} \cong .5745.$ The optimal cost is

$$J^* = J(u^*) \cong 1.7715 .$$

Table (C1.1) compares $\hat{J}^N$ and J* for each of the two schemes AVE and SPLINE. The example is somewhat typical in that (as one might expect from theoretical investigations - both methods are **basically** first order, but the estimates for SPLINE indicate that one should expect slightly faster convergence for this scheme) SPLINE converges faster than AVE. Note that the error for SPLINE at N = 4 is less than the error for AVE at N = 32.

Table (CL.2) contains the CPU times for each run of the conjugate-gradient algorithm. As shown, the time to make each run is very reasonable for both schemes, although SPLINE requires slightly more time per run. These times were typical for all the scalar examples.

Tables (C1.3) and (C1.4) compare the controls $\hat{u}_A^N$ and $\hat{u}_S^N$ to u* for N = 4, 8, 16, 32. We observe that SPLINE provides a better approximation to u* than AVE. It is interesting to note that SPLINE is not as monotone in its convergence as AVE.

AVE			SPLINE		
N	$\hat{J}^N$	$|\hat{J}^N - J*|$	N	$\hat{J}^N$	$|\hat{J}^N - J*|$
4	1.72491	.0466	4	1.77320	.0017
8	1.74750	.0240	8	1.77179	.00029
16	1.75939	.0121	16	1.77164	.00014
32	1.76551	.00599	32	1.77159	.00009
$J* = 1.7715$			$J* = 1.7715$		

TABLE C1.1

AVE		SPLINE	
N	CPU Sec	N	CPU Sec
4	23.8	4	30.6
8	28.9	8	39.1
16	39.1	16	59.5
32	56.1	32	93.5

TABLE C1.2

		AVE			
time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$	$u*$
0.00	-2.0050	-2.0067	-2.0075	-2.0086	-2.0108
0.25	-1.7340	-1.7386	-1.7400	-1.7405	-1.7450
0.50	-1.4994	-1.5069	-1.5092	-1.5092	-1.5081
0.75	-1.2962	-1.3064	-1.3112	-1.3124	-1.3106
1.00	-1.1197	-1.1317	-1.1402	-1.1448	-1.1500
1.25	-0.9662	- .9774	- .9877	- .9955	-1.0054
1.50	- .8335	- .8397	- .8463	- .8527	- .8618
1.75	- .7214	- .7201	- .7170	- .7151	- .7181
2.00	- .6324	- .6272	- .6179	- .6082	- .5745
2.25	- .5708	- .5720	- .5708	- .5706	- .5745
2.50	- .5385	- .5527	- .5623	- .5683	- .5745
2.75	- .5291	- .5504	- .5621	- .5683	- .5745
3.00	- .5286	- .5504	- .5621	- .5683	- .5745

TABLE C1.3

	SPLINE				
time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$	u^*
0.0	-2.0177	-2.0109	-2.0109	-2.0109	-2.0108
0.25	-1.7447	-1.7439	-1.7413	-1.7416	-1.7415
0.50	-1.5053	-1.5070	-1.5081	-1.5084	-1.5081
0.75	-1.3083	-1.3094	-1.3105	-1.3106	-1.3106
1.00	-1.1506	-1.1501	-1.1495	-1.1491	-1.1500
1.25	-1.0085	-1.0027	-1.0061	-1.0054	-1.0054
1.50	- .8591	- .8648	- .8629	- .8615	- .8618
1.75	- .7123	- .7147	- .7199	- .7178	- .7181
2.00	- .6095	- .5928	- .5844	- .5805	- .5745
2.25	- .5759	- .5746	- .5749	- .5746	- .5745
2.50	- .5816	- .5750	- .5748	- .5746	- .5745
2.75	- .5694	- .5774	- .5745	- .5746	- .5745
3.00	- .5149	- .5418	- .5574	- .5658	- .5745

TABLE C1.4

EXAMPLE C2

In this control problem, which is the same as Example 3B of [6], we take as our system the scalar equation

$$\dot{x}(t) = \frac{\pi}{2} x(t-1) + u(t) , \quad 0 \le t \le 2$$

with initial data

$$x(s) \equiv 1 , \qquad\qquad -1 \le s \le 0 .$$

The payoff is given by

$$J(u) = \tfrac{1}{2}[x(2)]^2 + \tfrac{1}{2} \int_0^2 [u(s)]^2 ds .$$

The optimal control is given by (see page 56 in [6])

$$u^*(t) \cong \begin{cases} \delta[(\pi/2)(1-t) + 1] , & 0 \le t \le 1 , \\ \\ \delta & , & 1 \le t \le 2 , \end{cases}$$

where $\delta \cong -.9967$ (see page 12 of [6]) and the optimal cost is

$$J^* = J(u^*) \cong 2.6787 .$$

Table (C2.1) summarizes the convergence properties of J^N to J^* and compares the AVE and SPLINE schemes. Again we observe an improvement by using the SPLINE scheme. Tables (C2.2) and (C2.3) contain the control values for AVE and SPLINE. A graphical comparison of $\hat{u}_A$, $\hat{u}_s$ and u^* is presented in Figure C2.1.

	AVE			SPLINE	
N	$\hat{J}^N$	$|\hat{J}^N - J*|$	N	$\hat{J}^N$	$|J^N - J*|$
4	2.6765	.0022	4	2.6827	.0040
8	2.6891	.0104	8	2.6801	.0014
16	2.6894	.0107	16	2.6792	.0005
32	2.6864	.0077	32	2.6790	.0003
	$J* = 2.6787$			$J* = 2.6787$	

TABLE C2.1

AVE

time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$	$u*$
0.00	-2.6371	-2.6122	-2.5962	-2.5834	-2.5623
0.25	-2.1679	-2.1568	-2.1583	-2.1629	-2.1709
0.50	--1.7783	-1.7653	-1.7645	-1.7703	-1.7795
0.75	-1.4593	-1.4324	-1.4086	-1.3933	-1.3881
1.00	-1.2112	-1.1762	-1.1363	-1.1005	- .9967
1.25	-1.0410	-1.0240	-1.0069	- .9975	- .9967
1.50	- .9521	- .9709	- .9837	- .9910	- .9967
1.75	- .9263	- .9646	- .9831	- .9910	- .9967
2.00	- .9248	- .9645	- .9831	- .9910	- .9967

TABLE C2.2

SPLINE

time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$	$u*$
0.00	-2.5732	-2.5679	-2.5643	-2.5626	-2.5623
0.25	-2.1750	-2.1652	-2.1725	-2.1708	-2.1709
0.50	-1.7649	-1.7858	-1.7822	-1.7789	-1.7795
0.75	-1.3669	-1.3778	-1.3927	-1.3871	-1.3881
1.00	-1.0907	-1.0465	-1.0236	-1.0129	- .9967
1.25	- .9986	- .9966	- .9976	- .9968	- .9967
1.50	-1.0102	- .9987	- .9974	- .9968	- .9967
1.75	- .9904	-1.0028	- .9968	- .9968	- .9967
2.00	- .8955	- .9409	- .9672	- .9815	- .9967

TABLE C2.3
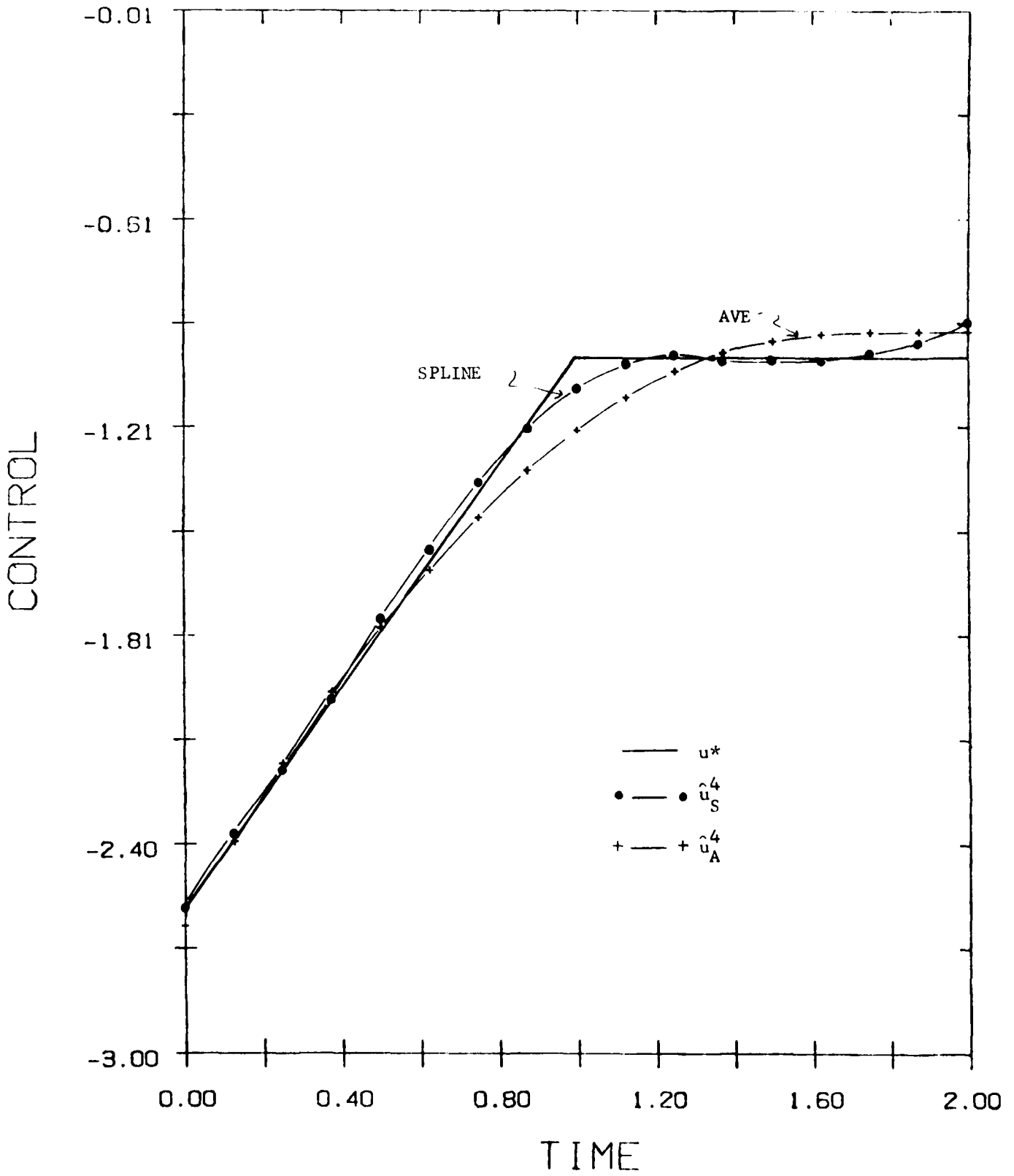
FIGURE C2.1

EXAMPLE C3

In this example the system is two dimensional and is described by

$$\frac{d}{dt} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1(t-1) \\ x_2(t-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t),$$

where $0 \le t \le 2$ and the initial data is chosen as

$$\begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} \equiv \begin{bmatrix} 10 \\ 0 \end{bmatrix} \quad , \; 1 \le s \le 0.$$

The cost functional is given by

$$J(u) = \tfrac{1}{2} [x_1(2)]^2 + \tfrac{1}{2} \int_0^2 [u(s)]^2 \, ds .$$

This system is equivalent to the second order equation (see Example 5 in [6])

$$\ddot{y}(t) + \dot{y}(t-1) = u(t),$$

where $x_1(t) = y(t)$ and $x_2(t) = \dot{y}(t)$. The optimal control is given by (see pages 18 and 66 of [6])

$$u^*(t) = \begin{cases} (\delta/2)(3 - t^2) & , \quad 0 \le t \le 1, \\ \\ \delta(2 - t) & , \quad 1 \le t \le 2, \end{cases}$$

where $\delta \doteq -3.1915$, and the optimal cost is

$$J* = J(u*) \cong 15.9574 \ .$$

Table C3.1 illustrates the convergence of $\hat{J}^N$ to J* for AVE and SPLINE. It is interesting to compare the CPU times (Table C3.2) for this two dimensional example with Table (C1.2) for the scalar problem, Example C1. Note that the total CPU time increased only a few seconds. Tables (C3.3) and (C3.4) compare $\hat{u}_A^N$ and $\hat{u}_S^N$ to u*. Note that $\hat{u}_S^{32}$ is almost identical (to 3 places) to u*!

AVE			SPLINE		
$\underline{N}$	$\hat{J}^N$	$\lvert\hat{J}^N - J*\rvert$	$\underline{N}$	$\hat{J}^N$	$\lvert\hat{J}^N - J*\rvert$
4	17.3450	1.3876	4	16.0149	.0575
8	16.7215	.7641	8	15.9721	.0147
16	16.3604	.4030	16	15.9618	.0044
32	16.1649	.2075	32	15.9594	.0020
J* = 15.9574			J* = 15.9574		

TABLE C3.1

AVE		SPLINE	
$\underline{N}$	CPU Sec	$\underline{N}$	CPU Sec
4	25.5	4	28.9
8	28.9	8	40.8
16	34.0	16	62.9
32	54.4	32	105.4

TABLE C3.2

AVE				
$\underline{time}$	$\underline{\hat{u}^4}$	$\underline{\hat{u}^8}$	$\underline{\hat{u}^{16}}$	$\underline{\hat{u}^{32}}$	$\underline{u^*}$
0.00	-4.8886	-4.8486	-4.8210	-4.8044	-4.7872
0.25	-4.7369	-4.7186	-4.7063	-4.6982	-4.6875
0.50	-4.4217	-4.4065	-4.3988	-4.3947	-4.3883
0.75	-3.9406	-3.9124	-3.8976	-3.8919	-3.8896
1.00	-3.3083	-3.2590	-3.2275	-3.2099	-3.1915
1.25	-2.5562	-2.4926	-2.4499	-2.4239	-2.3936
1.50	-1.7279	-1.6711	-1.6359	-1.6164	-1.5957
1.75	- .8670	- .8360	- .8180	- .8082	- .7979
2.00	0.00	0.00	0.00	0.00	0.00

TABLE C3.3

		SPLINE			
time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$	u^*
0.00	-4.8629	-4.8047	-4.7918	-4.7885	-4.7872
0.25	-4.7443	-4.7024	-4.6908	-4.6884	-4.6875
0.50	-4.4094	-4.3940	-4.3906	-4.3889	-4.3883
0.75	-3.8599	-3.8852	-3.8880	-3.8893	-3.8896
1.00	-3.1497	-3.1794	-3.1883	-3.1906	-3.1915
1.25	-2.3598	-2.3835	-2.3909	-2.3930	-2.3936
1.50	-1.5463	-1.5829	-1.5925	-1.5950	-1.5957
1.75	- .7381	- .7834	- .7942	- .7970	- .7979
2.00	0.00	0.00	0.00	0.00	0.00

TABLE C3.4

EXAMPLE C4

Consider now the three dimensional system

$$\frac{d}{dt} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(t-1) \\ x_2(t-1) \\ x_3(t-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t) ,$$

on the interval $0 \le t \le 3$. We choose as initial function the constant vector

$$\begin{bmatrix} x_1(s) \\ x_2(s) \\ x_3(s) \end{bmatrix} \equiv \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} , \quad -1 \le s \le 0 ,$$

and the cost is defined by

$$J(u) = (.05) \left[x_1(3) \right]^2 + \frac{1}{2} \int_0^3 \left[u(s) \right]^2 \, ds .$$

This is the same problem as Example 6 in the report [6]. The optimal control is found to be (see page 70 of [6])

$$u^*(t) = \begin{cases} \delta (1 - t)^2 , & 0 \le t \le 1 , \\ 0.0 , & 1 \le t \le 3 , \end{cases}$$

where $\delta = -.4975$, and

$$J^* = 4.9751$$

A numerical summary for this problem is presented in Table (C4.1)-(C4.4) and Figure (C4.1). The total CPU time is quite reasonable, even for this 3 dimensional system. Note that for

SPLINE it was necessary to carry $\hat{J}^N$ out to 6 places in order to compute the difference $|\hat{J}^N - J*|$.

AVE			SPLINE		
N	$\hat{J}^N$	$|\hat{J}^N - J*|$	N	$\hat{J}^N$	$|\hat{J}^N - J*|$
4	4.9216	.0595	4	4.9754	.0003
8	4.9504	.0248	8	4.9752	.0001
16	4.9635	.0116	16	4.9752	.0001
32	4.9696	.0055	32	4.9752	.0001
	$J* = 4.9751$			$J* = 4.9751$	

TABLE C4.1

AVE		SPLINE	
N	CPU Sec	N	CPU Sec
4	40.8	4	54.4
8	51.0	8	81.6
16	69.7	16	129.2
32	107.1	32	221.0

TABLE C4.2

AVE					
time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$	u^*
0.00	-.7201	-.6159	-.5577	-.5274	-.4975
0.25	-.4914	-.3958	-.3402	-.3102	-.2798
0.50	-.3153	-.2326	-.1830	-.1548	-.1244
0.75	-.1871	-.1212	-.0819	-.0591	-.0311
1.00	-.1002	-.0537	-.0281	-.0145	0.0
1.25	-.0469	-.0190	-.0065	-.0017	0.0
1.50	-.0183	-.0049	-.0009	-.0001	0.0
1.75	-.0055	-.0008	0.00	0.00	0.0
2.00	-.0011	-.0001	0.00	0.00	0.0
2.25	-.0001	0.00	0.00	0.00	0.0
2.50	0.00	0.00	0.00	0.00	0.0
2.75	0.00	0.00	0.00	0.00	0.0
3.00	0.00	0.00	0.00	0.00	0.0

TABLE C4.3

			SPLINE		
time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$	u^*
0.00	-.4932	-.4961	-.4966	-.4968	-.4975
0.25	-.2767	-.2791	-.2793	-.2794	-.2798
0.50	-.1256	-.1245	-.1242	-.1242	-.1244
0.75	-.0397	-.0322	-.0313	-.0311	-.0311
1.00	-.0052	-.0016	-.0005	-.0002	0.00
1.25	-.0015	-.0003	0.00	0.00	0.00
1.50	-.0005	0.00	0.00	0.00	0.00
1.75	-.0001	0.00	0.00	0.00	0.00
2.00	0.00	0.00	0.00	0.00	0.00
2.25	0.00	0.00	0.00	0.00	0.00
2.50	0.00	0.00	0.00	0.00	0.00
2.75	0.00	0.00	0.00	0.00	0.00
3.00	0.00	0.00	0.00	0.00	0.00

TABLE C4.4

FIGURE C4.1

EXAMPLE C5

This example involves a two dimensional system with two
controls. The equation is (see Example 7 in [6]) given by

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} x_1(t-1) \\ x_2(t-1) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \,,$$

where $0 \leq t \leq 2$, and the initial data is defined by

$$\begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} \equiv \begin{bmatrix} 1 \\ 1 \end{bmatrix} \,, \quad -1 \leq s \leq 0 \,.$$

The cost function is taken as

$$J(u) = \tfrac{1}{2}\{[x_1(2)]^2 + [x_2(2)]^2\} + \tfrac{1}{2}\int_0^2 \{[u_1(s)]^2 + [u_2(s)]^2\}\, ds \,.$$

The optimal control is (see pages 25 and 73 of [6]) found to
be

$$u^*(t) = \begin{bmatrix} u_1^*(t) \\ u_2^*(t) \end{bmatrix} \,,$$

where

$$u_1^*(t) = \begin{cases} \mu + \delta(1-t) & , \quad 0 \leq t \leq 1, \\ \mu & , \quad 1 \leq t \leq 2, \end{cases}$$

and

$$u_2^*(t) = \delta \qquad\quad , \quad 0 \leq t \leq 2,$$

with

$$\mu \;=\; -0.1880 \quad \text{and} \quad \delta \;=\; -0.8718 \;.$$

The optimal payoff has value

$$J^* \;=\; 1.4017 \;.$$

The results for this problem are summarized in Tables (C5.1)-(C5.5), and Figures (C5.1) - (C5.2).

AVE			SPLINE		
$\underline{N}$	$\hat{J}^N$	$|\hat{J}^N - J^*|$	N	J^N	$|J^N - J^*|$
4	1.3620	.0397	4	1.4072	.0055
8	1.3839	.0178	8	1.4035	.0018
16	1.3940	.0077	16	1.4022	.0005
32	1.3983	.0034	32	1.4019	.0001
$J^* \;=\; 1.4017$			$J^* \;=\; 1.4017$		

TABLE C5.1

AVE - u_1				
time	$\hat{u}_1^4$	$\hat{u}_1^8$	$\hat{u}_1^{16}$	$\hat{u}_1^{32}$	u_1^*
0.00	-1.0250	-1.0383	-1.0489	-1.0546	-1.0598
0.25	- .8239	- .8261	- .8320	- .8368	- .8418
0.50	- .6342	- .6219	- .6181	- .6195	- .6239
0.75	- .4639	- .4373	- .4188	- .4087	- .4060
1.00	- .3236	- .2919	- .2656	- .2449	- .1880
1.25	- .2248	- .2051	- .1927	- .1872	- .1880
1.50	- .1725	- .1748	- .1797	- .1836	- .1880
1.75	- .1574	- .1712	- .1793	- .1836	- .1880
2.00	- .1564	- .1711	- .1793	- .1836	- .1880

TABLE C5.2

AVE - u_2				
time	$\hat{u}_2^4$	$\hat{u}_2^8$	$\hat{u}_2^{16}$	$\hat{u}_2^{32}$	u_2^*
0.00	- .8558	- .8655	- .8695	- .8709	- .8718
0.25	- .8558	- .8655	- .8695	- .8709	- .8718
0.50					
0.75					
1.00					
1.25					
1.50					
1.75	- .8558	- .8655	- .8695	- .8709	- .8718
2.00	- .8558	- .8655	- .8695	- .8709	- .8718

TABLE C5.3

SPLINE - u_1				
time	$\hat{u}_1^4$	$\hat{u}_1^8$	$\hat{u}_1^{16}$	$\hat{u}_1^{32}$	u_1^*
0.00	-1.0531	-1.0641	-1.0607	-1.0597	-1.0598
0.25	- .8340	- .8398	- .8415	- .8411	- .8418
0.50	- .5980	- .6227	- .6226	- .6224	- .6239
0.75	- .3732	- .3931	- .4045	- .4035	- .4060
1.00	- .2202	- .2066	- .1978	- .1945	- .1880
1.25	- .1677	- .1781	- .1832	- .1856	- .1880
1.50	- .1716	- .1800	·· .1831	- .1856	- .1880
1.75	- .1698	- .1807	- .1830	- .1856	- .1880
2.00	- .1533	- .1695	- .1776	- .1827	- .1880

TABLE C5.4

SPLINE - u_2^*				
time	$\hat{u}_2^4$	$\hat{u}_2^8$	$\hat{u}_2^{16}$	$\hat{u}_2^{32}$	u_2^*
0.00	-.8959	-.8882	-.8783	-.8744	-.8718
0.25	-.9021	-.8864	-.8787	-.8745	-.8718
0.50	-.9067	-.8916	-.8790	-.8741	
0.75	-.9031	-.8893	-.8786	-.8745	
1.00	-.8956	-.8875	-.8782	-.8744	
1.25	-.8963	-.8881	-.8782	-.8744	
1.50	-.9022	-.8866	-.8781	-.8744	
1.75	-.8810	-.8904	-.8776	-.8744	-.8718
2.00	-.7972	-.8355	-.8516	-.8610	-.8718

TABLE C5.5
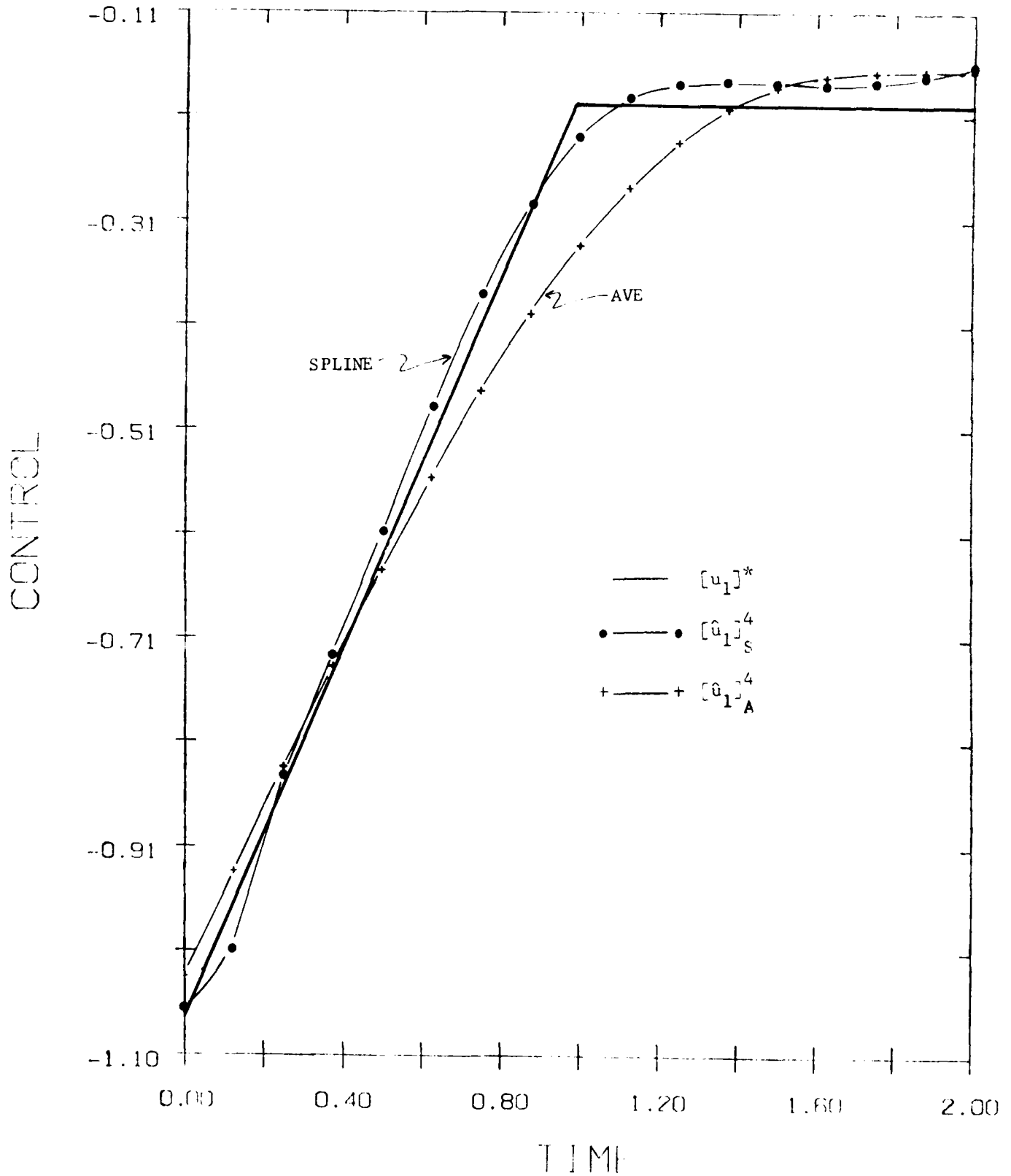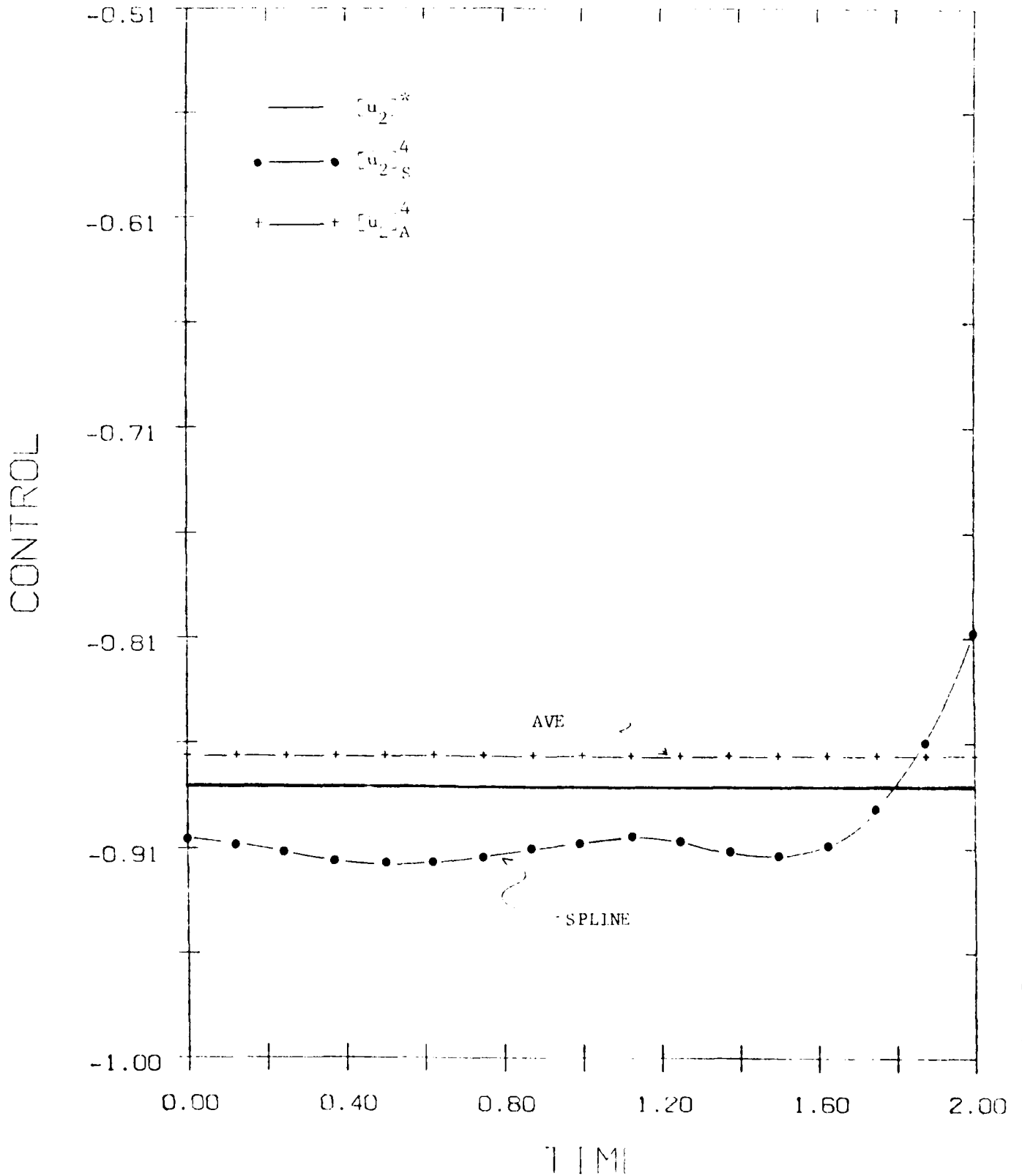
FIGURE C5.1

FIGURE C5.2

EXAMPLE C6

We consider a control problem with system given by the scalar equation

$$\dot{x}(t) = x(t) + x(t-1) + u(t), \quad 0 \le t \le 2,$$

and initial data

$$x(s) = 1, \quad -1 \le s \le 0.$$

The cost function is defined by

$$J(u) = \frac{3}{2}[x(2)]^2 + \frac{1}{2}\int_0^2 [u(s)]^2 \, ds.$$

The optimal control (see Example 4 on pages 14 and 63 in [6]) is given by

$$u^*(t) = \begin{cases} \delta[e^{2-t} + (1-t)e^{1-t}], & 0 \le t \le 1, \\ \delta e^{2-t}, & 1 \le t \le 2, \end{cases}$$

where $\delta = -.3932$ and

$$J^* = 3.1017.$$

The numerical results for this example are summarized in Tables (C6.1)-(C6.3). Observe again that the SPLINE scheme gives better approximations to the payoff and control than AVE.

AVE			SPLINE		
$\underline{N}$	$\hat{J}^N$	$\|\hat{J}^N-J*\|$	$\underline{N}$	$\hat{J}^N$	$\|\hat{J}^N-J*\|$
4	3.0084	.0933	4	3.1108	.0092
8	3.0554	.0462	8	3.1047	.0031
16	3.0797	.0219	16	3.1030	.0013
32	3.0915	.0101	32	3.1026	.0009
$J* \quad = \quad 3.1017$			$J* \quad = \quad 3.1017$		

TABLE C6.1

AVE					
$\underline{time}$	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$	$\underline{u*}$
0.00	-3.9668	-3.9702	-3.9726	-3.9734	-3.9742
0.25	-2.8512	-2.8662	-2.8754	-2.8807	-2.8870
0.50	-2.0497	-2.0672	-2.0759	-2.0808	-2.0863
0.75	-1.4759	-1.4917	-1.4964	-1.4968	-1.4986
1.00	-1.0684	-1.0843	-1.0883	-1.0868	-1.0688
1.25	- .7835	- .8042	- .8154	- .8221	- .8324
1.50	- .5876	- .6136	- .6298	- .6389	- .6483
1.75	- .4517	- .4765	- .4904	- .4975	- .5049
2.00	- .3515	- .3711	- .3819	- .3875	- .3932

TABLE C6.2

	SPLINE				
time	$\tilde{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$	$u^\star$
0.00	-3.9717	-3.9737	-3.9735	-3.9733	-3.9742
0.25	-2.8895	-2.8854	-2.8870	-2.8864	-2.8870
0.50	-2.0873	-2.0894	-2.0871	2.0858	-2.0863
0.75	-1.5001	-1.4968	-1.4998	-1.4981	-1.4986
1.00	-1.0949	-1.0821	-1.0757	-1.0728	-1.0688
1.25	- .8330	- .8326	- .8326	- .8323	- .8324
1.50	- .6535	- .6488	- .6484	- .6482	- .6483
1.75	- .5012	- .5070	- .5048	- .5048	- .5049
2.00	- .3590	- .3727	- .3819	- .3872	- .3932

TABLE C6.3

EXAMPLE C7

This example is an interesting two dimensional problem for which the SPLINE scheme is clearly superior to the AVE scheme. It is the same as Example 9 in [6]. The equation is

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} x_1(t-1) \\ x_2(t-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) ,$$

where $0 \leq t \leq 2$, and the initial condition is given by

$$\begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} \equiv \begin{bmatrix} 10 \\ 0 \end{bmatrix} , \quad -1 \leq s \leq 0 .$$

The equation is the vector formulation of the second order scalar equation

$$\ddot{y}(t) + \dot{y}(t - 1) + y(t) = u(t) ,$$

which describes an harmonic oscillator with delayed damping. The cost function is given by

$$J(u) = 5 \left[x_1(2) \right]^2 + \tfrac{1}{2} \int_0^2 \left[u(s) \right]^2 ds .$$

The optimal control (see pages 31 and 81 in ⌐ 6 ⌐) is given by

$$u^*(t) = \begin{cases} \delta \sin (2 - t) + \frac{1}{2}(1-t) \sin (t - 1) , & 0 \leq t \leq 1 , \\[2ex] \delta \sin (2 - t) & , 1 \leq t \leq 2 , \end{cases}$$

where $c = 2.5599$. The optimal cost is

$$J* = 3.3991 .$$

This example demonstrates just how much improvement one can obtain by using the SPLINE scheme in place of AVE.

In order to check the convergence rates of the two algorithms, we assume that

$$e_N = | \hat{J}^N - J* | = K_1 (1/N)^\beta$$

and

$$\mathcal{F}_N = \| \hat{u}^N - u* \|_{L_2} = K_2 (1/N)^\beta ,$$

where K_1, K_2 and β are constants. The <u>convergence rate</u>, β, can be used to compare the two schemes. For example, an algorithm with $\beta = 2.0$ provides faster convergence than an algorithm with $\beta = 1.0$. Solving the above equations for β, we find that

$$\beta = \frac{\ell n(e_N/e_{2N})}{\ell n\, 2} = \frac{\ell n(\mathcal{F}_N/\mathcal{F}_{2N})}{\ell n\, 2} .$$

Consequently, β can be estimated from the numerical results. Table C7.1 indicates the computed value of β for $\hat{J}^N \to J*$. Note that β is approximately one for AVE and two for SPLINE. The values of $\| \hat{u}^N - u* \|_{L_2}$ used in Table C7.4 were estimated by using a simple Euler scheme for the integration. Again, the convergence rate β is approximately one for AVE and two for SPLINE.

AVE				SPLINE			
$\underline{N}$	$\underline{\hat{J}^N}$	$\underline{|\hat{J}^N - J*|}$	$\underline{\beta}$	$\underline{N}$	$\underline{\hat{J}^N}$	$\underline{|\hat{J}^N - J*|}$	$\underline{\beta}$
4	2.1515	1.2475	.78	4	3.5354	.1363	1.94
8	2.6711	.7280	.88	8	3.4345	.0354	1.91
16	3.0035	.3956	.94	16	3.4085	.0094	1.75
32	3.1929	.2062	---	32	3.4019	.0028	----
$J* = 3.3991$				$J* = 3.3991$			

TABLE C7.1

AVE					
$\underline{time}$	$\underline{\hat{u}^4}$	$\underline{\hat{u}^8}$	$\underline{\hat{u}^{16}}$	$\underline{\hat{u}^{32}}$	$\underline{u*}$
0.00	1.0403	1.1386	1.1931	1.2212	1.2506
0.25	1.4574	1.6371	1.7451	1.8038	1.8645
0.50	1.7277	1.9522	2.0898	2.1664	2.2467
0.75	1.8163	2.0447	2.1839	2.2628	2.3501
1.00	1.7136	1.9110	2.0259	2.0882	2.1541
1.25	1.4369	1.5839	1.6644	1.7052	1.7449
1.50	1.0259	1.1209	1.1727	1.1997	1.2273
1.75	.5313	.5788	.6052	.6191	.6333
2.00	0.00	0.00	0.00	0.00	0.00

TABLE C7.2

SPLINE				
time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$	u^*
0.00	1.3669	1.2786	1.2579	1.2527	1.2506
0.25	1.9770	1.8937	1.8717	1.8666	1.8645
0.50	2.3287	2.2683	2.2530	2.2485	2.2467
0.75	2.3805	2.3611	2.3527	2.3511	2.3501
1.00	2.1609	2.1552	2.1545	2.1543	2.1541
1.25	1.7446	1.7440	1.7447	1.7451	1.7449
1.50	1.2018	1.2210	1.2258	1.2271	1.2273
1.75	.5902	.6232	.6308	.6328	.6333
2.00	0.00	0.00	0.00	0.00	0.00

TABLE C7.3

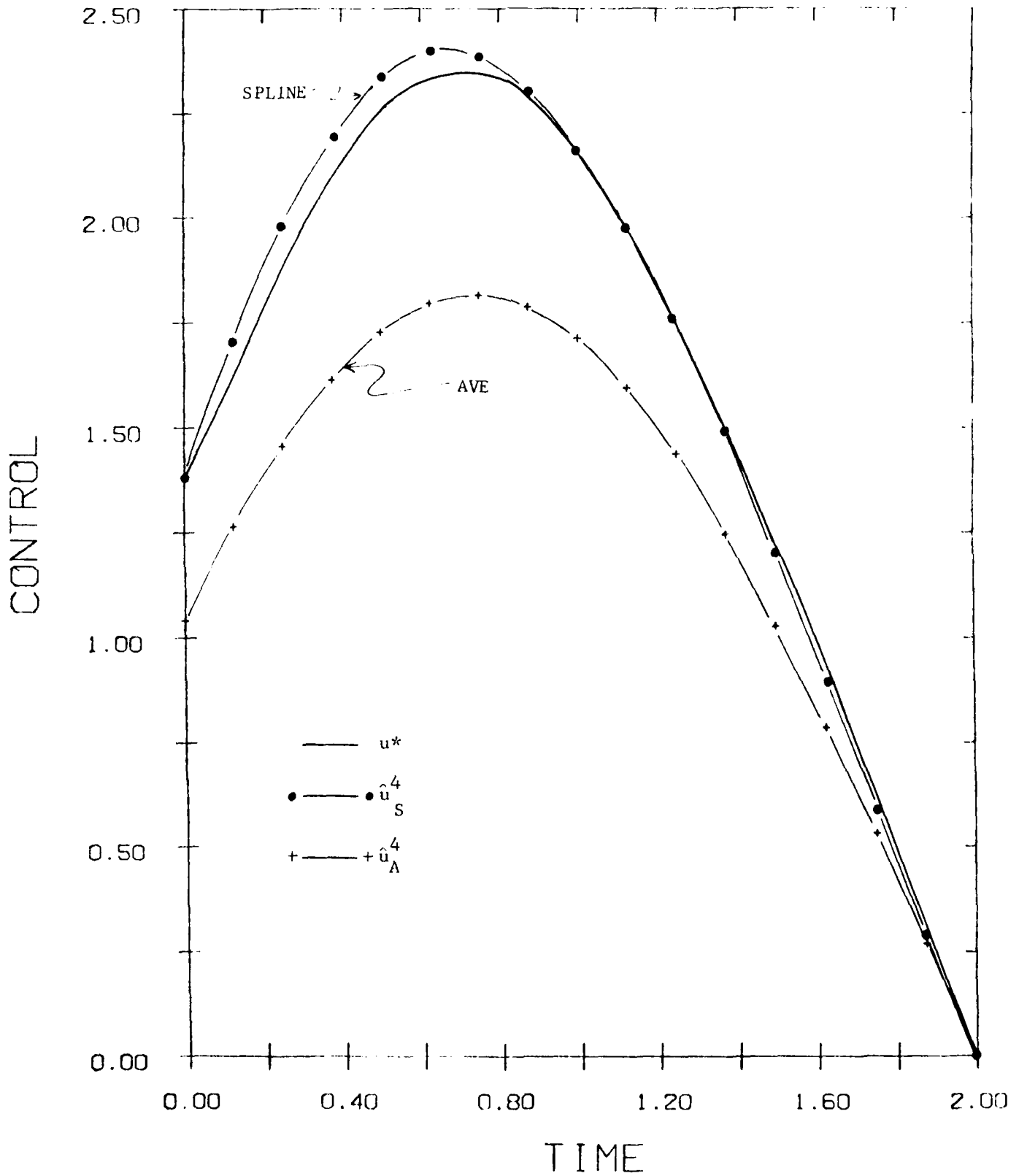| AVE			SPLINE		
N	$\|\hat{u}^N - u^*\|_{L_2}$	β	N	$\|\hat{u}^N - u^*\|_{L_2}$	β
4	.5650	..95	4	.0954	1.96
8	.2931	.92	8	.0245	1.95
16	.1553	.97	16	.0063	1.75
32	.0795	---	32	.0018	----

TABLE C7.4

FIGURE C7.1

EXAMPLE C8

Except for the initial data this problem is the same as Example 10 in [6]. The system is described by the two dimensional equation

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} x_1(t-1) \\ x_2(t-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u(t) ,$$

where $0 \le t \le 2$, and initial data given by

$$\begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \end{bmatrix}, \quad -1 \le s \le 0 .$$

The system is the vector formulation of the second order scalar equation

$$\ddot{y}(t) + \dot{y}(t) + y(t-1) = u(t) .$$

The payoff is taken to be

$$J(u) = \tfrac{1}{2}\{[x_1(2), x_2(2)]\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_1(2) \\ x_2(2) \end{bmatrix}\} + \tfrac{1}{2}\int_0^2 [u(s)]^2\, ds$$

$$= \tfrac{1}{2}\{[x_1(2)]^2 + [x_2(2)]^2 + \tfrac{1}{2}\int_0^2 [u(s)]^2\, ds .$$

Proceeding exactly as in [6] with $\varphi = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ replaced by $\varphi = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$ (see pages 36-45), one finds that the optimal control is given by

$$u^*(t) = \begin{cases} (\mu-\delta)e^{t-2} + [2\mu-3\delta-(\mu-\delta)t]e^{t-1} + \delta(t+2)-\mu, \ 0 \le t \le 1, \\ \\ (\mu-\delta)e^{t-2} + \delta \qquad\qquad\qquad\qquad\qquad\quad, \ 1 \le t \le 2, \end{cases}$$

where $\delta \simeq -.2593$ and $\mu \simeq 5.2262$. The resulting optimal cost is

$$J^* \simeq 19.7479 .$$

The convergence of $\hat{J}^N$ to J^* is summarized in Table (C8.1). Again, the convergence rates (i.e.,β) agree with the expected theoretical values. The convergence of $\hat{u}^N$ to u^* is described by Tables (C8.2)-(C8.4) and plots of $\hat{u}_A^4$, $\hat{u}_S^4$ and u^* are given in Figure (C8.1)

	AVE				SPLINE		
$\underline{N}$	$\underline{\hat{J}^N}$	$\lvert\underline{\hat{J}^N-J^*}\rvert$	$\underline{\beta}$	$\underline{N}$	$\underline{\hat{J}^N}$	$\lvert\underline{\hat{J}^N-J^*}\rvert$	$\underline{\beta}$
4	17.9646	1.7832	.87	4	19.9843	.2364	2.39
8	18.7745	.9733	.95	8	19.7929	.0450	1.72
16	19.2439	.5039	.99	16	19.7616	.0137	1.48
32	19.4935	.2543	---	32	19.7528	.0049	----
	$J^* = 19.7479$				$J^* = 19.7479$		

TABLE C8.1

AVE				
time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$	u^*
0.00	-.9029	-.8782	-.8712	-.8701	-.8710
0.25	-.3412	-.2688	-.2307	-.2133	-.1993
0.50	.2526	.3612	.4276	.4636	.4975
0.75	.8643	.9879	1.0680	1.1170	1.1745
1.00	1.4926	1.6054	1.6739	1.7132	1.7587
1.25	2.1585	2.2469	2.2927	2.3139	2.3319
1.50	2.9139	2.9886	3.0277	3.0476	3.0678
1.75	3.8368	3.9187	3.9645	3.9883	4.0128
2.00	5.0122	5.1115	5.1673	5.1962	5.2262

TABLE C8.2

SPLINE				
time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$	u^*
0.00	-.8775	-.9746	-.9297	-.9010	-.8710
0.25	-.1303	-.2995	-.2441	-.2231	-.1993
0.50	.6421	.4472	.4667	.4785	.4975
0.75	1.3224	1.1445	1.1595	1.1662	1.1745
1.00	1.9007	1.7394	1.7560	1.7575	1.7587
1.25	2.5345	2.3528	2.3485	2.3399	2.3319
1.50	3.3597	3.1217	3.1068	3.0871	3.0678
1.75	4.2533	4.1425	4.0785	4.0463	4.0128
2.00	4.9126	5.0619	5.1643	5.1955	5.2262

TABLE C8.3

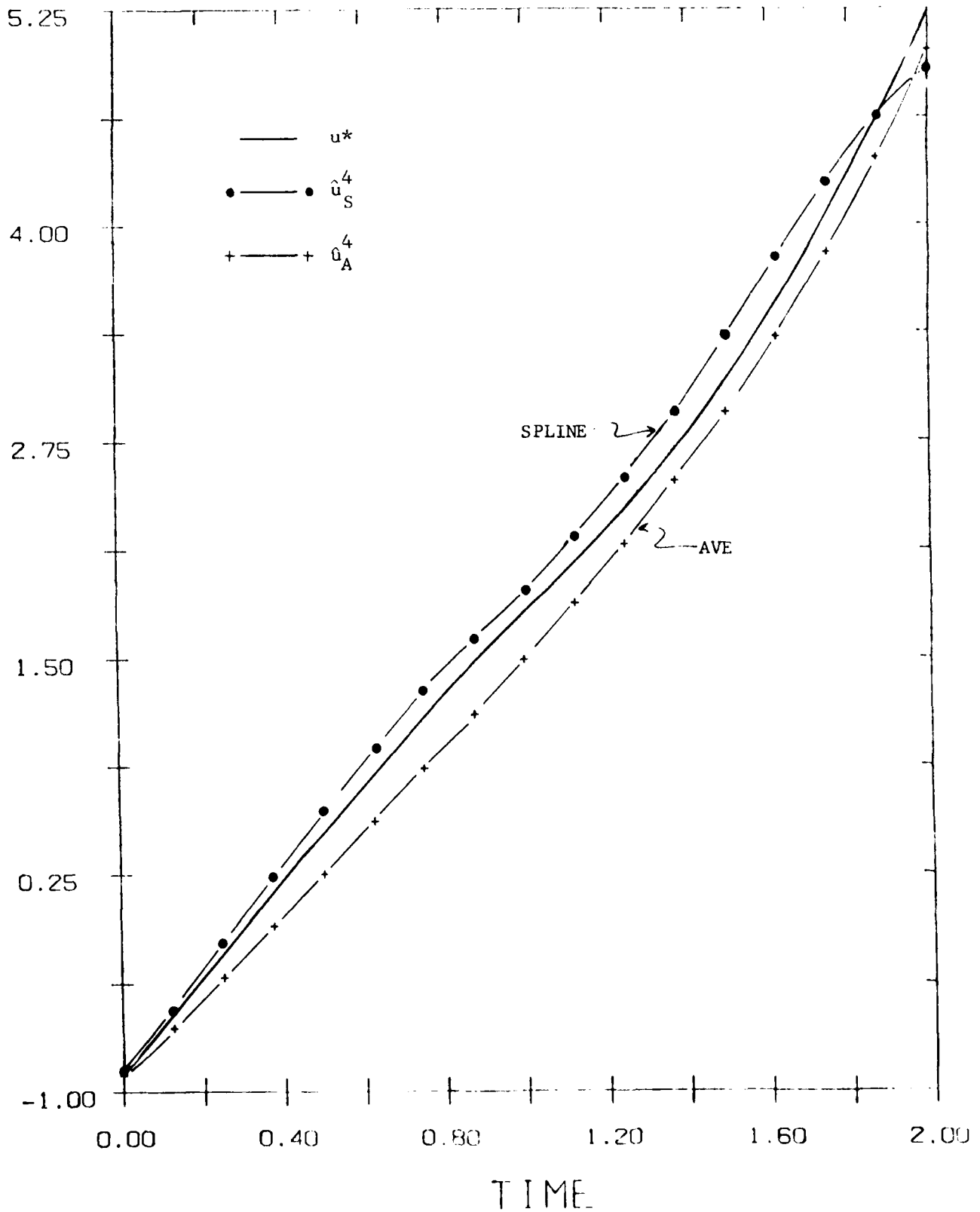| AVE			SPLINE		
$\underline{N}$	$\|\underline{\hat{u}^N - u^*}\|_{L_2}$	$\underline{\beta}$	$\underline{N}$	$\|\underline{\hat{u}^N - u^*}\|_{L_2}$	$\underline{\beta}$
4	.0944	1.69	4	.0875	2.52
8	.0293	1.81	8	.0153	2.26
16	.0083	1.90	16	.0032	1.84
32	.0022	----	32	.0009	----

TABLE C8.4

FIGURE C8.1

EXAMPLE C9

This example is the same as Example 13 in [6] (with
$\gamma_1 = \gamma_2 = 1$, see page 97 of [6]). The system is governed by the
equation

$$
\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} x_1(t-1) \\ x_2(t-1) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix},
$$

with $0 \leq t \leq 2$ and initial data

$$
\begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \end{bmatrix}, \quad -1 \leq s \leq 0 .
$$

The cost function is given by

$$
J = \tfrac{1}{2}\{[x_1(2)]^2 + [x_2(2)]^2\} + \tfrac{1}{2}\int_0^2 \{[u_1(s)]^2 + [u_2(s)]^2\}\, ds .
$$

Although the optimal control and optimal cost have not been

computed, the numerical results given in the following tables

are similar to the previous examples. Again it appears

that SPLINE provides improved convergence properties over AVE.

AVE			SPLINE		
$\underline{N}$	$\hat{J}^N$	$\|\hat{J}^{2N}-J^N\|$	$\underline{N}$	$\hat{J}^N$	$\|\hat{J}^{2N}-J^N\|$
4	16.2927	.7615	4	18.0819	.1814
8	17.0542	.4084	8	17.9005	.0348
16	17.4626	.2014	16	17.8657	.0090
32	17.6640	-----	32	17.8567	-----

TABLE C9.1

AVE - u_1				
time	$\hat{u}_1^4$	$\hat{u}_1^8$	$\hat{u}_1^{16}$	$\hat{u}_1^{32}$
0.00	-2.3842	-2.5442	-2.6432	-2.6920
0.25	-1.9485	-2.0417	-2.1157	-2.1601
0.50	-1.4369	-1.4407	-1.4639	-1.4947
0.75	- .8927	- .7962	- .7311	- .7000
1.00	- .3798	- .2166	- .0816	.0247
1.25	.0235	.1658	.2578	.3042
1.50	.2562	.3092	.3221	.3224
1.75	.3284	.3271	.3239	.3224
2.00	.3331	.3273	.3239	.3224

TABLE C9.2

AVE - u_2				
time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$
0.00	-.3729	-.3578	-.3541	-.3540
0.25	.1412	.1975	.2273	.2407
0.50	.6665	.7536	.8068	.8354
0.75	1.1900	1.2891	1.3527	1.3912
1.00	1.7101	1.7994	1.8512	1.8798
1.25	2.2453	2.3141	2.3453	2.3573
1.50	2.8403	2.8999	2.9258	2.9364
1.75	3.5615	3.6321	3.6650	3.6788
2.00	4.4788	4.5707	4.6139	4.6321

TABLE C9.3

SPLINE - u_1				
time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$
0.00	-2.7993	-2.7564	-2.7430	-2.7366
0.25	-2.2589	-2.2041	-2.2079	-2.2025
0.50	-1.4849	-1.5456	-1.5454	-1.5400
0.75	- .5920	- .6683	- .7219	- .7100
1.00	.0951	.1992	.2532	.2783
1.25	.3615	.3467	.3292	.3256
1.50	.3542	.3351	.3295	.3256
1.75	.3336	.3370	.3294	.3256
2.00	.3040	.3162	.3196	.3206

TABLE C9.4

SPLINE - u_2				
time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$
0.00	-.3499	-.3416	-.3488	-.3520
0.25	.3107	.2604	.2622	.2555
0.50	.9737	.9025	.8753	.8649
0.75	1.5378	1.4784	1.4520	1.4428
1.00	1.9986	1.9456	1.9266	1.9185
1.25	2.4949	2.4122	2.3821	2.3716
1.50	3.1407	2.9946	2.9646	2.9528
1.75	3.8248	3.7720	3.7109	3.6988
2.00	4.2899	4.4311	4.5281	4.5842

TABLE C9.5

EXAMPLE C10

This is a nonlinear optimal control problem. Although the basic idea has been developed only for linear systems, much of the theory has been extended to a general class of nonlinear examples (see [1]). The system is given by the equation

$$\dot{x}(t) = x(t)\sin[x(t)] + x(t-1) + u(t), \quad 0 \le t \le 2 .$$

with initial data

$$x(s) \equiv 10 , \quad -1 \le s \le 0 .$$

The cost function is

$$J = \tfrac{1}{2} [x(2)]^2 + \tfrac{1}{2} \int_0^2 \{[x(s)]^2 + [u(s)]^2\} \, ds .$$

Values of $\hat{J}^N$ are given in Table (C10.1). This problem is such that J^* is relative "flat". Consequently, the values of $\hat{J}^N$ and the controls $\hat{u}^N$ changed very little as $N \to +\infty$. Since the optimal control u^* is not known analytically and $\hat{u}^N$ were essentially the same for $N \ge 4$, we did not give tables for $\hat{u}^N$.

AVE			SPLINE		
N	$\hat{J}^N$	$\|\hat{J}^{2N}-J^N\|$	N	$\hat{J}^N$	$\|\hat{J}^{2N}-J^N\|$
4	162.020	.0010	4	162.113	.0720
8	162.019	.0010	8	162.041	.0300
16	162.018	.0030	16	162.011	.0080
32	162.015	-----	32	162.003	-----

TABLE C10.1

EXAMPLE C11

This is a nonlinear example with the same dynamics as EXAMPLE C10, but different initial condition. The system is again given by

$$\dot{x}(t) = x(t)\sin[x(t)] + x(t-1) + u(t) , \quad 0 \le t \le 2 ,$$

with initial data

$$x(s) = \varphi(s) , \quad -1 \le s \le 0 ,$$

where

$$\varphi(s) = \begin{cases} 10(s + 1) , & -1 \le s \le -\tfrac{1}{2} , \\ \\ -10s , & -\tfrac{1}{2} \le s \le 0 . \end{cases}$$

The cost functional is given by

$$J(u) = \tfrac{1}{2}[x(s)]^2 + \tfrac{1}{2}\int_0^2 \{[x(s)]^2 + [u(s)]^2\} \, ds .$$

This nonlinear problem is more interesting than Example C10. Although the optimal control is not known analytically, the numerical runs indicate that the sequence $\{\hat{u}^N\}$ is "converging" to an optimal control. If one applies the maximum principle to the nonlinear control problem, there are two boundary conditions that the optimal "state" and multiplier must satisfy. Using these boundary conditions as a check for the

approximating optimal control problem, we found that the $N = 16$

SPLINE procedure produced final values of the "state" and

"multipliers" that most nearly matched the boundary conditions.

In view of this fact and the convergence pattern illustrated in

Tables C11.1 - C11.3, it is reasonable to believe that the $N = 16$

SPLINE run gives a good estimate of the optimal control.

Figure C11.1 compares plots of $\hat{u}_S^4$, $\hat{u}_A^4$ with $\hat{u}_S^{16}$. The plots of

$\hat{u}_S^N$ for $N = 8$, 16 and 32 are almost identical.

AVE			SPLINE		
N	$\hat{J}^N$	$\lvert \hat{J}^{2N} - J^N \rvert$	N	J^N	$\lvert J^{2N} - J^N \rvert$
4	1.9919	.1845	4	2.5406	.0179
8	2.1764	.1341	8	2.5227	.0013
16	2.3105	.0907	16	2.5240	.0010
32	2.4012	-----	32	2.5230	-----

TABLE C11.1

AVE				
time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$
0.00	-2.3013	-2.3082	-2.3073	-2.3033
0.25	-2.0915	-2.1804	-2.2375	-2.2706
0.50	-1.8241	-1.9856	-2.1114	-2.1972
0.75	-1.4673	-1.5752	-1.6470	-1.6906
1.00	-1.1190	-1.1447	-1.1417	-1.1317
1.25	- .8349	- .8173	- .7891	- .7697
1.50	- .6208	- .5966	- .5761	.5669
1.75	- .4580	- .4400	- .4321	- .4329
2.00	- .3233	- .3125	- .3098	-..3142

TABLE C11.2

SPLINE				
time	$\hat{u}^4$	$\hat{u}^8$	$\hat{u}^{16}$	$\hat{u}^{32}$
0.00	-2.331	-2.3024	-2.3017	-2.2967
0.25	-2.3707	-2.3230	-2.3156	-2.3101
0.50	-2.2820	-2.3034	-2.3106	-2.3126
0.75	-1.6921	-1.7294	-1.7404	-1.7424
1.00	-1.0787	-1.1002	-1.1041	-1.1037
1.25	- .7160	.7427	- .7462	- .7497
1.50	- .5727	- .5644	- .5588	- .5624
1.75	- .4978	- .4591	- .4502	- .4451
2.00	- .3960	- .3496	- .3559	- .3265

TABLE C11.3
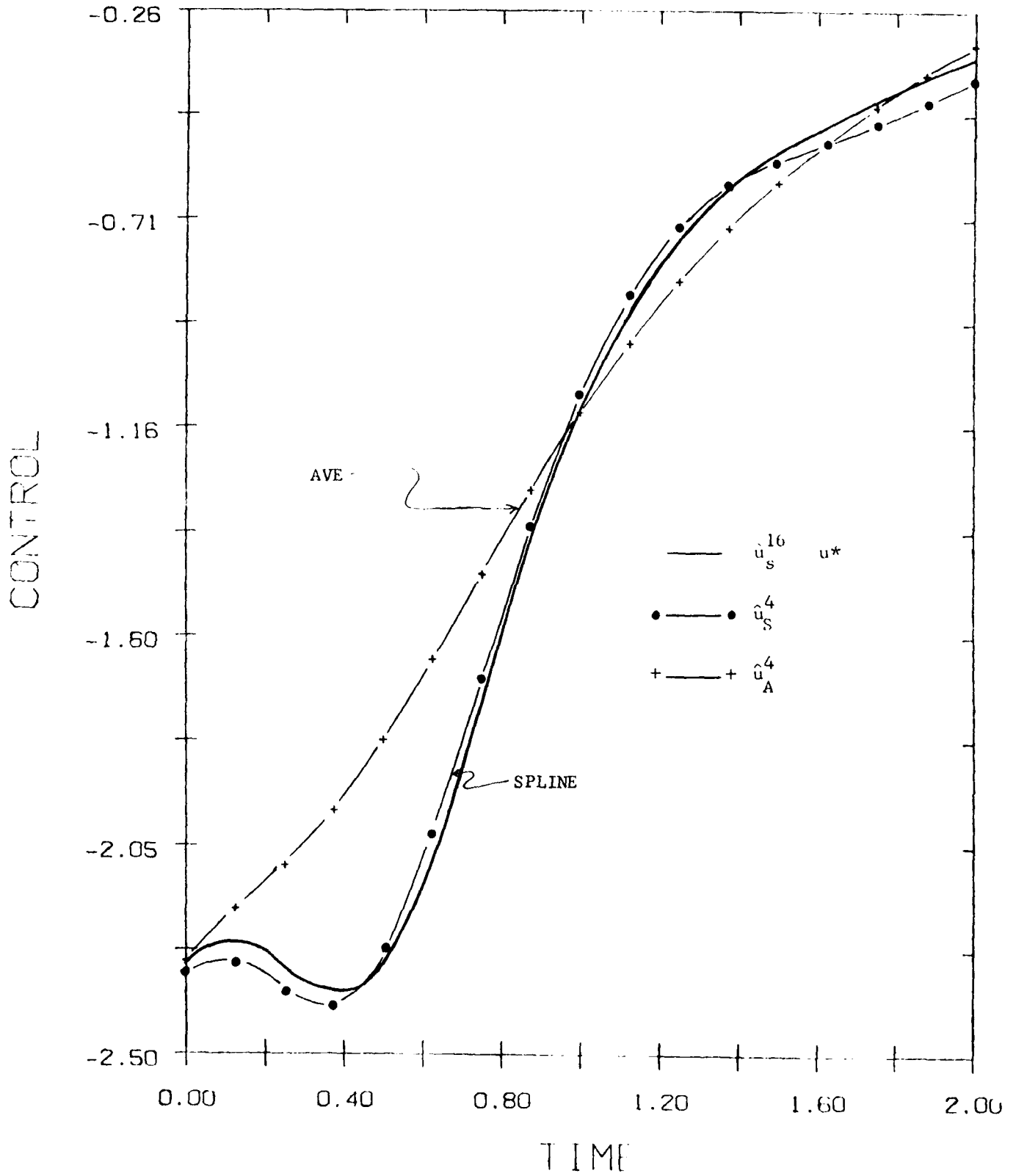
FIGURE C11.1

Summary Remarks

In examples C7 and C8 we presented some estimates for the rates of convergence of $J^N \to J^*$ and $u^N \to u$. In these cases we found that the AVE scheme provided essentially linear convergence, while SPLINE convergence was essentially quadratic. In view of some known theoretical facts these results are not unexpected. Specifically, in [5] it is shown that for fixed control and sufficiently re-stricted initial data, we have $z^N \to z$ is $O(1/N^\beta)$ where $\beta = 1$ for AVE and $\beta = 2$ for SPLINE. For the linear regulator problem (both OC and OCN) the optimal control can be generated by state feedback. If one assumes that the feedback "parameters" for OCN also converge to the feedback "parameters" for OC (see Delfour [10]) like $(1/N^\beta)$, then it follows that $\hat{u}^N \to u^*$ convergence is $O(1/N^\beta)$. Additionally, elementary calculations then reveal that one should have $\hat{J}^N \to J^*$ is also $O(1/N^\beta)$. For the first-order (piecewise linear) spline based method SPLINE, the results presented here, taken with other numerical experiments that we have performed and reported else-where (see [8]), are strong evidence that the method SPLINE is essentially second order ($\beta = 2$) when used with regulator-type optimal control problems involving linear delay systems. (This is not too surprising when one reviews the literature on finite-element methods and investigates such phenomena as super convergence for "coercive problems" or "superconvergence at nodes".) The AVE scheme, on the other hand, appears to be at best only <u>first</u> order ($\beta = 1$)

in these problems.

As a further point of comparison we note that SPLINE yields a better approximation for a given value of N in almost every example that we have considered (here and elsewhere). Observe that in many of the control examples (C1 - C4, C6 - C8) the results for SPLINE at $N = 4$ are better than those obtained with AVE for $N = 32$.

While the SPLINE scheme is slightly more tedious to impliment and takes a little more computer time because of the matrix systems that must be solved (see the Remark at the end of §3), all the evidence would seem to imply that SPLINE is a superior method to AVE in control problems of the type we consider here.

For control problems with nonlinear systems, the numerical findings to date are not as dramatic or conclusive (in part perhaps because analytic solutions are not available). In addition to the two examples (C10, C11) reported here, numerical studies with other nonlinear systems tend to support the conjecture that SPLINE will generally be as good as or better than AVE for nonlinear problems.

References

[1]   H. T. Banks, Approximation of nonlinear functional differential
      equation control systems, J. Opt. Theory Appl., 1979, to appear.

[2]   H. T. Banks and J. A. Burns, An abstract framework for approxi-
      mate solutions to optimal control problems governed by heredi-
      tary systems, in Proceedings, International Conference on
      Differential Equations, (Univ. So. Calif., Sept., 1974), H. A.
      Antosiewicz, ed., Academic Press, New York, 1975, pp. 10-25.

[3]   H. T. Banks and J. A. Burns, Hereditary control problems:
      numerical methods based on averaging approximations, SIAM
      J. Control and Optimization, 16 (1978), 169-208.

[4]   H. T. Banks and J. A. Burns, Approximation techniques for
      control systems with delays, Proc. Int'l Conference on Methods
      of Mathematical Programming, Polish Scientific Publishers,
      Warsaw, 1977, to appear.

[5]   H. T. Banks, J. A. Burns and E. M. Cliff, Parameter estimation
      and identification for systems with delays, November 1979, to
      appear.

[6]   H. T. Banks, J. A. Burns, E. M. Cliff, and P. R. Thrift,
      Numerical solutions of hereditary control problems via an
      approximation technique, Brown University LCDS Tech. Rep.
      75-6, Providence, R.I., October, 1975.

[7]   H. T. Banks and G. M. Groome, Jr., Convergence Theorems for
      parameter estimation by quasilinearization, J. Math. Anal. Appl.,
      42 (1973), 91-109.

[8]   H. T. Banks and F. Kappel, Spline approximations for functional
      differential equations, J. Diff. Eq., to appear.

[9]   J. A. Burns and E. M. Cliff, Parameter identification for
      hereditary systems, Final Technical Report on Grant AFFDL-77-
      3221A, February, 1979, AFFDL Technical Memorandum, to appear.

[10] M. C. Delfour, The linear quadratic optimal control problem
      for hereditary differential systems: theory and numerical
      solutions, Applied Mathematics and Opt., 3 (1977), 101-162.

[11] J. Hale, Theory of Functional Differential Equations, Springer-
      Verlag, New York, 1977.

DATE
FILMED

4-80

DTIC